

**Baber, Patrick**

Performancesteigerung von Webanwendungen  
auf Softwareebene am Beispiel des  
Content Management Systems TYPO3

**– Bachelorarbeit –**

Hochschule Mittweida – University of Applied Science (FH)

**Baber, Patrick**

Performancesteigerung von Webanwendungen  
auf Softwareebene am Beispiel des  
Content Management Systems TYPO3

**– eingereicht als Bachelorarbeit –**

Hochschule Mittweida – University of Applied Science (FH)

Erstprüfer  
Prof. Dr.-Ing. Robert J. Wierzbicki

Zweitprüfer  
Wanle Bohoui

**Bibliographische Beschreibung:**

Baber, Patrick:

Performancesteigerung von Webanwendungen auf Softwareebene am Beispiel des Content Management Systems TYPO3 – 2009 – 62 S.

Mittweida, Hochschule Mittweida (FH), Fachbereich Medien, Bachelorarbeit, 2009

**Referat:**

Ziel dieser Bachelorarbeit ist es, praxistaugliche Ansätze zur Optimierung des Content Management Systems TYPO3 zu entwickeln. Die erarbeiteten Ansätze sollen das CMS hinsichtlich seiner Performance verbessern, um es bei hochfrequentierten Webseiten einsetzen zu können. Diese Arbeit beschäftigt sich mit den am häufigsten verwendeten Komponenten zum Betreiben einer Webseite. Es wird auf das Betriebssystem Debian GNU/Linux, den Webserver Apache 2, die Skriptsprache PHP und die Datenbank MySQL eingegangen, die als Basis der TYPO3-Konfiguration dienen. Viele dieser Ausführungen sind dennoch in abstrahierter Form auch auf andere Systeme anwendbar. Themenspezifisches Know-How wird für das Verständnis der Optimierungsarbeiten auf allen Ebenen vermittelt.

## Inhaltsverzeichnis

I	Abbildungsverzeichnis .....	6
II	Tabellenverzeichnis .....	6
III	Abkürzungsverzeichnis .....	7
1	Problemstellung .....	7
2	Zielsetzung .....	10
3	Ebenen der Performance-Optimierung .....	13
3.1	Hardware .....	13
3.2	Internetanbindung .....	14
3.3	Betriebssystem .....	15
3.4	XAMPP/LAMPP .....	16
3.5	TYPO3 .....	17
4	Apache .....	18
4.1	Diagnose .....	18
4.1.1	Das Diagnose-Tool „ab“ .....	19
4.1.2	Das Diagnose-Modul „mod_info“ .....	20
4.1.3	Das Diagnose-Modul „mod_status“ .....	21
4.2	Multi-Processing-Module .....	21
4.2.1	prefork .....	21
4.2.2	worker .....	22
4.2.3	Ausblick auf Apache 2.3/2.4 .....	22
4.3	Einbindung von Modulen .....	23
4.4	Komprimierung mit mod_deflate .....	24
4.5	DNS-Lookups vermeiden .....	25
5	PHP .....	27
5.1	PHP-Version .....	28
5.2	Einbindung des PHP-Interpreters .....	29
5.2.1	mod_php .....	30
5.2.2	PHP als CGI via suPHP .....	30
5.2.3	PHP als CGI via mod_fcgid .....	31
5.2.4	Vergleich .....	32
5.3	php.ini .....	34
5.3.1	always_populate_raw_post_data .....	34
5.3.2	max_execution_time .....	35
5.3.3	expose_php .....	35

5.3.4	memory_limit.....	36
5.3.5	register_globals.....	36
5.4	PHP-Beschleuniger.....	37
5.4.1	OpCode Cache.....	37
5.4.2	OpCode Optimizer.....	38
5.4.3	Vergleich.....	38
5.4.4	eAccelerator.....	40
<b>6</b>	<b>MySQL.....</b>	<b>42</b>
6.1	Diagnose .....	42
6.1.1	Slow Query Log.....	42
6.1.2	EXPLAIN .....	43
6.1.3	SHOW STATUS.....	43
6.2	Konfiguration.....	44
6.2.1	key_buffer_size.....	44
6.2.2	Query Cache .....	45
6.2.3	table_open_cache.....	46
<b>7</b>	<b>TYPO3 .....</b>	<b>47</b>
7.1	Caching.....	47
7.1.1	Client.....	47
7.1.2	Server.....	48
7.1.2.1	Caching-Tabellen .....	48
7.1.2.2	Caching im Backend beeinflussen .....	49
7.1.2.3	Caching mit TypoScript beeinflussen.....	50
7.1.2.4	nc_staticfilecache .....	52
<b>8</b>	<b>Schluss.....</b>	<b>53</b>
<b>9</b>	<b>Literaturverzeichnis .....</b>	<b>55</b>
9.1	Bücher.....	55
9.2	Zeitschriften.....	55
9.3	Internetquellen.....	56
9.4	Internetquellen (Wikipedia).....	61
<b>10</b>	<b>Selbständigkeitserklärung.....</b>	<b>62</b>

## Abbildungsverzeichnis

Abbildung 1: Client-Server-Kommunikation.....	25
Abbildung 2: PHP 5 im Vergleich .....	29
Abbildung 3: Funktionsweise von PHP .....	29
Abbildung 4: Auswertung dynamischen Contents mit 500 KB Dateien.....	33
Abbildung 5: Speicherverbrauch mod_php vs. FastCGI.....	34
Abbildung 6: PHP-Beschleuniger im Vergleich.....	39
Abbildung 7: TYPO3-Cache im Backend beeinflussen .....	50
Abbildung 8: Cache im Backend löschen .....	51

## Tabellenverzeichnis

Tabelle 1: Caching-Tabellen einer TYPO3-Grundinstallation.....	49
--	----

## Abkürzungsverzeichnis

CGI: Common Gateway Interface

CMS: Content Management System

DB: Database

DSO: Dynamic Shared Object

FCGI: Fast Common Gateway Interface

MPM: Multi-Processing-Module

PHP: Personal Home Page Tools (vor Version 3.0), PHP: Hypertext Preprocessor  
(ab Version 3.0)

VHost: Virtueller Host

## 1 Problemstellung

Das Internet und die damit verbundene Online-Welt entwickeln sich rasant. Nicht zuletzt durch jüngste Entwicklungen des Web 2.0<sup>1</sup> wachsen damit die Anforderungen an eine Internetseite. Ursprünglich zur Beschaffung von Informationen als Alternative zum Fernsehen und den Printmedien, entwickelt sich die Branche jüngst mehr und mehr zur Bezugsquelle der ersten Wahl. Seither hat sich das Internet als ein Alltagsmedium für Information, Unterhaltung und Kommunikation etabliert. Webseiten wachsen zu aufwendigen Applikationen heran, wie die Social Networks<sup>2</sup> Facebook<sup>3</sup> und Twitter<sup>4</sup> eindrucksvoll beweisen. Über Webseiten wie Facebook unterhält man sich mit der Bekanntschaft aus Übersee, präsentiert Fotos des letzten Urlaubes und zeigt seinen Freunden auf welcher Veranstaltung man am Abend zu finden ist. Twitter geht einen leicht anderen Weg. Über die Plattform „zwitschern“, das heißt teilen aktuell rund 11,5 Mio. Nutzer ihren Freunden mit, was sie gerade tun.<sup>5</sup> Die wachsende Funktionalität, wie auch die enormen Zugriffszahlen erfordern aber zugleich eine entsprechende Technik, um die teils komplex generierten Inhalte auch unter Lastspitzen auszuliefern und auf der Seite des Nutzers darzustellen. Für die Übertragung komplexer Inhalte, wie z. B. Grafiken und Videos, werden Breitbandanbindungen nötig. Bei der kürzlich durch die Bundeskanzlerin Angela Merkel ausgesprochenen Forcierung des Breitbandausbaus<sup>6</sup> werden nun die wettbewerbsrechtlichen Rahmenbedingungen durch die Bundesnetzagentur entwickelt. Diese sollen dafür sorgen, dass sich Großunternehmen und Global Player wie die Deutsche Telekom und Vodafone am Ausbau der Infrastruktur beteiligen und so bis Ende 2010 die Versorgung aller deutschen Haushalte mit einem Breitbandanschluss sicherstellen.

Neben der schnellen Übertragung mittels Breitbandanbindung sind für moderne Webseiten viele weitere Technologien nötig. Man unterscheidet dabei zwei Hinsichten. Zum einen gibt es die Seite des Website-Users (Client) und der bereitstellenden Seite des Website-Betreibers (Server)<sup>7</sup>. Auf der Client-Seite kommt für die Darstellung aktueller Webseiten oft Javascript<sup>8</sup>, eine Skriptsprache, zum Einsatz. Das haben die Browserentwickler bereits erkannt und optimieren aus diesem Grund

<sup>1</sup> Siehe O'Reilly: „What is Web 2.0“, 30.09.2005, <http://oreilly.com/web2/archive/what-is-web-20.html>, 01.07.2009

<sup>2</sup> Vgl. Schaumann: „Neue Social Networks und neue Formen der Kollaboration“, o.J., <http://philipps-welt.info/social%20networks.htm>, 03.07.2009

<sup>3</sup> URL: <http://www.facebook.com/>

<sup>4</sup> URL: <http://www.twitter.com/>

<sup>5</sup> Vgl. Cheng/Evans: „Inside Twitter – An In-Depth Look Inside the Twitter World“, 06/2009 <http://www.sysomos.com/insidetwitter/>, 01.07.2009

<sup>6</sup> Vgl. Mansmann: „Kanzlerin will Breitbandausbau forcieren“, 01.02.2009 <http://www.heise.de/newsticker/Kanzlerin-will-Breitbandausbau-forcieren-/meldung/126696>, 02.07.2009

<sup>7</sup> Siehe: eTeaching.org: „Client-Server“, 09.08.2007, <http://www.e-teaching.org/technik/vernetzung/architektur/client-server/>, 03.07.2009

<sup>8</sup> Vgl. Schäfer/Strübig: „Einführung in JavaScript und DOM“, o.J., <http://de.selfhtml.org/javascript/intro.htm>, 03.07.2009



gezielt ihre Rendering Engines<sup>9</sup>, die für die Darstellung der Internetseite im Browser zuständig sind, um somit den wachsenden Anforderungsprofilen gerecht zu werden. Für die User-Ebene wird also einiges getan, um den rasanten Entwicklungen des Onlinemarktes gerecht zu werden. Auch die Server-Seite benötigt jedoch eine verlässliche und zudem performante Konfiguration zur Abdeckung der Bedürfnisse. Eine Internetseite muss rund um die Uhr erreichbar sein und dem Nutzer dürfen auch unter Lastspitzen keine horrenden Ladezeiten davon abhalten an die gesuchten Informationen zu gelangen.

Der Aufbau, sowie die Pflege und Wartung eines solch hochverfügbaren Servers fällt in den Kompetenzbereich des Administrators und ist damit klar von der Arbeit eines Web Developers getrennt, der sich mit der Entwicklung der Website beschäftigt. Nach meinen eigenen Erfahrungen schmelzen beide Berufsgruppen, um Kosten zu sparen, mehr und mehr zusammen. Dies ist besonders bei den vielschichtigen Optimierungsarbeiten zur Performancegewinnung bei Webseiten von Vorteil. Beispielsweise lässt sich auf diese Weise sowohl die Website, als auch die zur Bereitstellung der Inhalte nötige Technik, von nur einer Berufsgruppe optimieren. Denn es ist in vielen Fällen nicht notwendig, die angestaubte Hardware durch aktuelle Komponenten zu ersetzen. Vielmehr wird die Frage nach kostengünstigen Alternativen laut. Um solche Alternativen soll es in dieser Arbeit gehen.

Damit ist der Rahmen dieser komplexen Thematik skizziert. Im Folgenden wird mit der Zielsetzung erläutert, welche möglichen Antworten für die genannten Problemfelder beschrieben werden. Dabei wird wie folgt vorgegangen. Zuerst erfolgt eine Beschreibung der Softwarepakete, die innerhalb dieser speziellen Konfiguration Verwendung finden. Anschließend werden die Vor- und Nachteile dieser Lösungsversuche diskutiert.

---

<sup>9</sup> Vgl. Eicker, „Layout Engine“, o.J., <http://www.kleines-lexikon.de/w/l/layoutengine.shtml>, 04.07.2009

## 2 Zielsetzung

Für die Umsetzung einer komplexen Website stehen, neben der Möglichkeit eine Website von der Pike auf selbst zu programmieren, zahlreiche Open-Source-Lösungen<sup>10</sup> zur Verfügung. Sie kommen dem Entwickler dabei in unterschiedlichem Maße entgegen. Von einfachen Klassen<sup>11</sup>, die für gängige Probleme der Website-Entwicklung Lösungen liefern, bis hin zu schnell einsatzbereiten Content Management Systemen, die ein breites Spektrum an Funktionalität mitbringen. Im Verlauf dieser Arbeit soll gezeigt werden, wie solch ein System auch bei hochfrequentierten Webseiten eingesetzt werden kann.

Zu den am häufigsten eingesetzten Lösungen in diesem Bereich zählt TYPO3. Es gehört neben Applikationen, wie Joomla<sup>12</sup> und Drupal<sup>13</sup> zu den sogenannten Entwicklungsframeworks, die als eine Art Grundgerüst für die endgültige Website fungieren. Ein vordefiniertes Layout kann in ein solches System integriert werden, um es mit Inhalten, wie Texten und Bildern zu bestücken.

Das ursprünglich von Kasper Skårhøj entwickelte TYPO3, steht unter der General Public License<sup>14</sup> und ist damit frei verfügbar. Laut den Angaben der TYPO3 Association gehört das System mit rund 290.000 Installationen<sup>15</sup> zu den populärsten Content Management Systemen (CMS). Die auf der Skriptsprache PHP5 und der Datenbank MySQL basierende Plattform bildet das Rückgrat vieler kleiner und mittelständiger Unternehmenswebseiten mit mittlerem Besucheraufkommen.

Die nachfolgende Arbeit hat das Ziel, zu zeigen, dass TYPO3 auch für Internetpräsenzen mit hohem Besucheraufkommen eine sehr gute Basis bietet. Im Vordergrund steht dabei die Auseinandersetzung mit Ansätzen zur software-seitigen Optimierung. Neben dem CMS, bietet gerade die von TYPO3 benötigte Server-Software viel Raum zur Verbesserung. Die Open-Source-Plattform wurde für eine Vielzahl von Konfigurationen entwickelt. Auf sämtliche Varianten einzugehen, würde den Rahmen dieser Arbeit bei Weitem sprengen. Aus diesem Grund konzentrieren sich die Ausführungen auf eine konkrete, die am häufigsten verwendete Konfiguration - dem LAMP-System<sup>16</sup>. Viele der beschriebenen Ansätze werden dennoch auf andere Systeme, in leicht abgewandelter Form, übertragbar sein.

<sup>10</sup> Vgl. Nix: „Was ist eigentlich Open Source?“, 01.01.2005, [http://www.contentmanager.de/magazin/artikel\\_843\\_was\\_ist\\_eigentlich\\_open\\_source.html](http://www.contentmanager.de/magazin/artikel_843_was_ist_eigentlich_open_source.html), 11.08.2009

<sup>11</sup> Siehe: Objektorientierte Programme, [http://de.wikipedia.org/wiki/Objektorientierte\\_Programmierung](http://de.wikipedia.org/wiki/Objektorientierte_Programmierung), 14.08.2009

<sup>12</sup> Vgl. Joomla Team: „What is Joomla?“, o.J., <http://www.joomla.org/about-joomla.html>, 11.08.2009

<sup>13</sup> Vgl. Buytaert: „About Drupal“, 03.07.2009, <http://drupal.org/about>, 08.07.2009

<sup>14</sup> Vgl. Free Software Foundation: „GNU General Public License“, 29.06.2007, <http://www.gnu.org/copyleft/gpl.html> 08.07.2009

<sup>15</sup> Vgl. TYPO3 Association: „TYPO3 Association“, o.J. <http://typo3.de/Facts-and-Figures.factsandfigures.0.html>, 09.07.2009

<sup>16</sup> Vgl. Dornfest: „LAMP Lighter: The Apache Toolbox“, 17.11.2000, <http://onlamp.com/pub/a/apache/2000/11/17/wrangler.html> 02.08.2009

Diese Bachelorarbeit befasst sich vornehmlich mit frei verfügbarer, sogenannter Open-Source-Software. Als Basis für die Maßnahmen zur Optimierung der Performance dient die in deutschen Unternehmen am häufigsten verwendete Linux-Distribution Debian.<sup>17</sup> Die beschriebenen Themengebiete dieser Arbeit sollten somit mühelos auf artverwandte Konfigurationen wie Ubuntu oder andere auf dem Linux-Kern basierende Betriebssysteme anwendbar sein. Die in dieser Arbeit behandelte Konfiguration umfasst neben dem Betriebssystem insgesamt vier Software-Pakete, die nun näher erläutert werden.

Für die Auslieferung der Website-Daten an den User ist ein Webserver nötig. In diesem Bereich erwies sich der Apache HTTP Server,<sup>18</sup> entwickelt von der Apache Software Foundation, als solide Basis. Die Software bietet eine gute Skalierbarkeit im Einsatz und ist wie alle folgenden Software-Pakete kostenlos verfügbar. Laut aktuellen Zahlen von Netcraft ist der Apache 2 Webserver mit knapp über 100 Mio. Webseiten und einem Marktanteil von rund 47% der Marktführer.<sup>19</sup> Der meist auf Linux-Distributionen zum Einsatz kommende Webserver dient also in vielen Systemen als Basis für die Client-Server-Kommunikation und wird daher auch in dieser wissenschaftlichen Arbeit zur Grundlage genommen.

TYPO3 basiert seit jeher, auf der beliebten Skriptsprache PHP. Die aktuellen Releases von TYPO3, ab der Version 4.2.0, benötigen zwingend die PHP-Version 5.2 oder höher.<sup>20</sup> Im Hinblick auf die Stabilität des Server-Systems wird grundsätzlich nur auf getestete und damit stabile Komponenten, gesetzt. Außer acht gelassen werden hingegen Software-Pakete, die sich noch im Frühstadium ihrer Entwicklung befinden.

Zur Speicherung der großen Datenmengen greift TYPO3 auf eine Datenbank zurück. Aufgrund der programmierten Abstraktionsschicht ist es möglich TYPO3 mit einer Vielzahl von unterschiedlichen Datenbankverwaltungssystemen, wie Oracle<sup>21</sup> oder PostgreSQL<sup>22</sup> zu betreiben. Die am häufigsten eingesetzte unkommerzielle Lösung ist hier jedoch MySQL<sup>23</sup>. Die Konfiguration dieser Software kann sehr aufwändig sein. Gerade bei hochfrequentierten Webseiten macht es Sinn mehrere MySQL-Server zu einem sogenannten Server-Cluster zu verbinden.

<sup>17</sup> Vgl. Diedrich: „Trendstudie Open Source – Eingesetzte Produkte“, 04.02.2009 <http://www.heise.de/open/Trendstudie-Open-Source-/artikel/126682/8>, 11.08.2009

<sup>18</sup> Vgl. Apache Software Foundation.: „What ist he Apache http Server Project?“, o.J., [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html), 11.08.2009

<sup>19</sup> Vgl. Netcraft: „July 2009 Web Server Survey“, 28.07.2009, [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html), 12.08.2009

<sup>20</sup> Vgl. Stucki: „Leaving PHP4 behind...“, 13.07.2007 <http://buzz.typo3.org/people/stucki/article/leaving-php4-behind/>, 18.07.2009

<sup>21</sup> Siehe: ORACLE Deutschland: „Warum Oracle?“, o.J., <http://www.oracle.com/lang/de/database/index.html>, 03.08.2009

<sup>22</sup> Siehe: PostgreSQL Global Development Group: „About“, o.J. <http://www.postgresql.org/about/>, 03.08.2009

<sup>23</sup> Siehe: MySQL AB: „MySQL AB Completes Record Year“, 30.01.2007 [http://www.mysql.com/news-and-events/generate-article.php?id=2007\\_02](http://www.mysql.com/news-and-events/generate-article.php?id=2007_02), 18.07.2009

Aufgrund der Komplexität des Themas soll an dieser Stelle jedoch nur auf die Ein-Server-Lösung eingegangen werden.

Mit diesen genannten Punkten und dem zuvor in der Problemstellung erarbeiteten Thema, ergibt sich folgendes Untersuchungsziel: Die Arbeit untersucht und diskutiert Ansätze der Performancesteigerung von Webanwendungen auf Softwareebene am Beispiel des Content Management Systems TYPO3.

Bevor die Thematik näher erläutert wird, sei noch erwähnt, dass für die Umsetzung der in dieser Bachelorarbeit entwickelten Ansätze grundlegende Linux-Kenntnisse von Nöten sind. Zudem wird ein Root-Zugang zum Server, auf dem die Maßnahmen zur Optimierung durchgeführt werden sollen, benötigt. Solides Wissen im Bereich der Webtechnologien, allen voran PHP und MySQL, sowie Know-How bei der Verwendung von Webservern, wie dem Apache wird vorausgesetzt.

### 3 Ebenen der Performance-Optimierung

Unabhängig um welche Applikation es geht, am Anfang einer jeden Optimierung steht das lokalisieren vermeidlicher Flaschenhälse. Für Online-Applikationen, wie TYPO3 gibt es fünf Ebenen, die relevant sind. Es ist wichtig diese zu kennen und in ihren Grundzügen zu verstehen, bevor man sich an die Diagnose wagt. Es handelt sich um die folgenden fünf Ebenen:

1. Hardware
2. Internetanbindung
3. Betriebssystem
4. XAMPP/LAMPP
5. TYPO3

Im Zuge dieser Arbeit wird gezielt auf die letzten beiden Ebenen eingegangen. Diese letzten beiden werden intensiver diskutiert, die vorhergehenden hingegen weniger ausführlich besprochen.

#### 3.1 Hardware

Die unterste Ebene zur Verbesserung der Leistung stellt die Hardware-Ebene dar. Abhängig von den Mindestanforderungen der Software wird eine damit korrelierte Leistung der Hardware vorausgesetzt. Die offizielle TYPO3-Seite gibt hier wenig Aufschluss über die vorausgesetzte Hardware. In den Mindestanforderungen heißt es:

„A normal webserver setup will do, with some modern CPU and at least 256 MB Ram. As with all database-driven applications, more RAM is advisable though.“<sup>24</sup>

Dieser noch sehr allgemein wirkende Hinweis lässt sich durch den sehr vielfältigen Einsatz von TYPO3 erklären. Das System kann für unterschiedliche Größen von Webseiten Anwendung finden. Welche Hardware-Komponenten benötigt werden kann so nicht pauschal gesagt werden, hängt es doch von zu vielen Faktoren ab.

Beim Aufruf einer TYPO3-Seite durchläuft die Generierung viele Ebenen der benötigten Server-Software. Die erforderliche Leistung ist daher abhängig von den eingesetzten Software-Komponenten. Diese arbeiten auf unterschiedliche Art und Weise und beanspruchen die Hardware daher mannigfaltig. Nach aktuellen Maßstäben lässt sich über TYPO3 allerdings sagen, dass die benötigte Hardware zum Betreiben einer moderaten TYPO3-Installation ein eher kleines Problem darstellt. Die meisten Hosting-Provider bieten mit ihrer angebotenen Server-Hardware eine solide Basis zum Betreiben einer TYPO3-Installation.<sup>25</sup> Erst durch die Verwendung von Erweiterungen für TYPO3 und durch ein hohes Besucheraufkommen steigen die

<sup>24</sup> TYPO3 Association: „System Requirements“, o.J., <http://typo3.org/about/system-requirements/>, 18.07.2009

<sup>25</sup> Vgl. Hauser/Neufeind (T3N): „Hosting für Profis“, 12/2008, S. 43 ff.

Anforderungen. Wer hier eine maßgeschneiderte Hardware-Lösung sucht, sollte sich seinen eigenen Rechner nach Belieben zusammensetzen, um ihn bei einem Provider, unter dem Stichwort „Colocation“<sup>26</sup>, an die verfügbare Infrastruktur anbinden zu lassen. Sollte darüberhinaus ein einzelner Server die Website-Anfragen unter Lastspitzen nicht mehr in ausreichendem Maße verarbeiten, können zusätzliche Rechner für die Lastenverteilung eingesetzt werden. Dies nennt man Load-Balancing und es verbindet mehrere Server zu einem Server-Cluster. Darauf wird in dieser Arbeit allerdings nicht weiter eingegangen. Um den schnellen Einstieg in TYPO3 zu finden, reicht also eine recht einfache Hardware aus, die im besten Fall für die wachsenden Ansprüche im Produktiveinsatz erweiterbar sein sollte. Bei leistungstärkerer Hardware sollte man das PreisLeistungsverhältnis allerdings nicht aus den Augen verlieren, da dieses Thema ein teures Unterfangen sein kann. Der Kauf stärkerer Hardware sollte als letzter Ausweg bei Performanceengpässen angesehen werden. Zuvor sollte ausreichend Optimierungsarbeit auf Software-Ebene geleistet werden.

Wie zu Beginn dieses Abschnitts erwähnt, besitzt TYPO3 grundsätzlich eine moderate Hardwareanforderung. Diese darf allerdings nur als Mindestanforderung angesehen werden, die für eine Internetpräsenz mit wenigen Besuchern ausreicht. Anders sieht es dagegen bei einer hochfrequentierten Website aus. Dort sollten jedoch zunächst Optimierungen an der Software vorgenommen werden, noch bevor neue Hardware gekauft wird.

## 3.2 Internetanbindung

Was nutzt es, wenn der eigene Server die Webseiten pfeilschnell generiert, sie aber aufgrund einer langsamen Internetanbindung nur sehr träge beim Nutzer ankommen? Dieser Umstand wird im Folgenden näher erläutert.

Die Geschwindigkeit der Übertragung ist abhängig von zwei Parteien: Der Website-anbietenden Server-Seite und der Client-Seite. Diese beiden Seiten stehen in wechselseitiger Kommunikation, die durch Protokolle standardisiert ist. Das Maß für eine solche Internetanbindung ist die Datenrate und wird gemessen in Dateneinheit pro Zeiteinheit.<sup>27</sup> Da die Datenrate des Internet-Nutzers, abhängig von dessen Provider ist und wir aus diesem Grund keinen Einfluss darauf haben, können wir nur auf der Server-Seite für eine schnelle Anbindung sorgen. Diese ist wiederum vom Provider und dessen Infrastruktur abhängig. Plakative Angaben, wie „6 GBit/s Außenanbindung“<sup>28</sup> helfen dabei nur in Maßen weiter, da es sich bei dieser Zahl um ein addiertes, theoretisches Maximum aller Datenraten des Rechenzentrums nach außen handelt. Der Server besitzt bei einem häufig eingesetzten Anschluss von

<sup>26</sup> Vgl. iX – Magazin für professionelle Informationstechnik: „Server-Housing“, o.J., <http://www.heise.de/ix/server-housing/>, 12.08.2009

<sup>27</sup> Siehe: „Datenübertragungsrate“, <http://de.wikipedia.org/wiki/Datenübertragungsrate>, 19.07.2009

<sup>28</sup> Vgl. Hostloco.com: „Häufige Fragen“, o.J., [http://www.hostloco.com/fragen\\_de.phtml](http://www.hostloco.com/fragen_de.phtml), 19.07.2009

100 MBit/s gar nicht die Möglichkeit diesen Wert auch nur annähernd auszunutzen. Dabei kommt erschwerend dazu, dass meist mehrere Server, in einem Serverschrank zusammengefasst, an eine 100 MBit/s-Leitung angeschlossen sind.<sup>29</sup> In solch einem Fall wird also noch nicht einmal der server-eigene Netzwerkanschluss vollständig ausgenutzt. Um diese Thematik allerdings noch auf die Spitze zu treiben, werden bei Webhosting-Paketen mehrere Kunden-Webseiten auf einem Server betrieben.<sup>30</sup> Neben der sinkenden Hardware-Leistung, leidet dabei natürlich auch der Datendurchsatz. Die Auslieferung der Website-Daten dauert länger. Es ist daher bei der Wahl des Server-Providers darauf zu achten, dass die Daten der Webseite in annehmbarer Zeit zum Benutzer übertragen werden, unter der Berücksichtigung von Reserven für Lastspitzen. Wer eine genaue Kalkulation wünscht, sollte Online-Tools zur Berechnung der Bandbreite, zu Rate ziehen.<sup>31</sup> In der Regel reicht aber eine Anbindung mit garantierten 100 MBit/s auch für größere Vorhaben aus.

### 3.3 Betriebssystem

Welches Betriebssystem ist das beste? Diese Frage lässt sich nicht so leicht beantworten. Sie erfordert ein genaues Anforderungsprofil, das es nun zu erarbeiten gilt.

Bei der Wahl des richtigen Betriebssystems spielen viele Faktoren eine Rolle. So muss etwa die benötigte Soft- und Hardware in ihrem Zusammenspiel reibungslos funktionieren. Wer sich beispielsweise für ein Unix-System entscheidet, kann gewisse Hardware aufgrund fehlender Treiber nur eingeschränkt oder vielleicht gar nicht nutzen. Möchte man bei der Software trotz alternativem Betriebssystem auf nichts verzichten, gibt es die Möglichkeit, mittels Virtualisierung<sup>32</sup> mehrere Betriebssysteme parallel zu nutzen. Eine Konfiguration mit einem Linux als *Wirtsystem* und Windows als *Gastsystem* wäre somit denkbar. Es liegt in der Natur der Sache, dass das auf dem Basissystem aufsetzende Betriebssystem nicht an die Leistung eines reinen Servers mit einem Betriebssystem und ohne Virtualisierung herankommt. Es ist also zu bedenken welches System als Grundlage dient und ob weitere Systeme benötigt werden.

Laut der offiziellen TYPO3-Seite stellt das CMS keine speziellen Anforderungen an das Betriebssystem.<sup>33</sup> Lediglich die darauf installierte Software, wie Webserver, Datenbank und Skriptinterpreter müssen vorhanden und funktionsfähig sein. Dennoch benötigt eine Server-Konfiguration eine grundsolide Basis, um eine Internetpräsenz rund um die Uhr betreiben zu können. Die Stabilität und Sicherheit spielt also eine große Rolle.

<sup>29</sup> Vgl. Neufeind (T3N): „Volle Kontrolle“, 06/2009, 52

<sup>30</sup> Vgl. ALL-INKL.COM: „Webhosting“, o.J., <http://all-inkl.com/index.php?open=uebersicht&sek=webhosting>, 20.07.2009

<sup>31</sup> Siehe: <http://www.bandbreiten.de/rechner/bandbreitenrechner.php>

<sup>32</sup> Vgl. Kolyshkin: „Virtualisierung hat viele Facetten“, 03/2008, 65 ff.

<sup>33</sup> Vgl. TYPO3 Association: „System Requirements“, o.J., <http://typo3.org/about/system-requirements/>, 18.07.2009



Die Linux-Distribution Debian verfolgt, wie eingangs bereits erwähnt, genau diese Strategie und benötigt aufgrund seiner schlanken Architektur wenige Ressourcen.

Natürlich ist die Wahl eines anwendungsspezifischen Betriebssystems ein breit gefächertes Feld, mit dem sich diese Arbeit nicht auseinander setzen kann und soll. Dennoch gibt es ein paar grundlegende Hilfestellungen, auf die im Folgenden kurz eingegangen wird. Zunächst sollten alle nicht benötigten Dienste deaktiviert werden. Eine Auflistung aller momentan laufenden Prozesse kann dabei helfen, die Ressourcenfresser zu lokalisieren.<sup>34</sup> Darüberhinaus sollte das Betriebssystem mit Aktualisierungen auf dem neusten Stand gehalten werden.<sup>35</sup> Diese bringen nicht nur Stabilität, sondern in einigen Fällen auch Leistung, wie zum Beispiel bei einem Kernel-Update auf Version 2.6.x und höher.<sup>36</sup> Dieses Upgrade steht für viele Linux-Distribution, wie auch Debian zur Verfügung.

### 3.4 XAMPP/LAMPP

Die nächste Ebene umfasst gleich mehrere Komponenten. Dabei handelt es sich um Komponenten, die man für die meisten Internetauftritte benötigt. So steht das Akronym XAMPP für **A**pache<sup>37</sup>, **M**ySQL<sup>38</sup>, **P**HP<sup>39</sup> und **P**erl<sup>40</sup>. Das vorangestellte **X** hingegen, ist als Leerstelle für ein beliebiges Betriebssystem zu lesen. Dieser Begriff wurde durch die Entwickler Apache Friends<sup>41</sup> geprägt. Abgesehen von Perl, benötigt TYPO3 genau diese Komponenten oder gängige Alternativen, wie z.B. der IIS als Ersatz für den Apache Webserver. Bei diesen Paketen handelt es sich um komplexe Software, die zunächst für den Einsatz facettenreicher Web-Applikationen ausgelegt sind. Im Produktiveinsatz stehen sie bei ordnungsgemäßer Konfiguration für hohe Stabilität und gute Performance. Speziell für die Verwendung mit TYPO3, lassen sich an diesen Komponenten einige Verbesserungen vornehmen. Diese werden in der vorliegenden Arbeit ab dem Kapitel 4 **Apache** näher erläutert.

<sup>34</sup> Vgl. Pro-Linux: „ps – Prozess-Status“, 22.11.1999 [http://www.pro-linux.de/t\\_shell/ps.html](http://www.pro-linux.de/t_shell/ps.html), 12.08.2009

<sup>35</sup> Vgl. Heise Zeitschriften Verlag GmbH & Co. KG: „heise Security“, o.J., <http://www.heise.de/security/>, 19.07.2009

<sup>36</sup> Vgl. Leemhuis: „Kernel-Log: 2.6.26-Entwicklung läuft schwungvoll an, Fehlerkorrektur für 2.6.24 und 2.4.36“, 21.04.2008, <http://www.heise.de/newsticker/Kernel-Log-2-6-26-Entwicklung-laeuft-schwungvoll-an-Fehlerkorrekturen-fuer-2-6-24-und-2-4-36-/meldung/106755>, 19.07.2009

<sup>37</sup> Siehe: Apache Software Foundation: „What is the Apache HTTP Server Project?“, o.J. [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html), 03.08.2009

<sup>38</sup> Siehe: MySQL AB: „Warum MySQL?“, o.J., <http://www.mysql.de/why-mysql/>, 03.08.2009

<sup>39</sup> Siehe: The PHP Group: „What is PHP?“, o.J., <http://www.php.net/>, 03.08.2009

<sup>40</sup> Siehe: The Perl Foundation: „About Perl“, o.J., <http://www.perl.org/about.html>, 03.08.2009

<sup>41</sup> Siehe: Apache Friends: „Willkommen bei Apache Friends“, <http://www.apachefriends.org/de/index.html>, 03.08.2009



### 3.5 TYPO3

Das populäre Content Management System steht aktuell in der Version 4.2.8<sup>42</sup> auf der offiziellen Webseite von TYPO3, zum Download bereit.<sup>43</sup> Das System stellt eine hoch komplexe Anwendung dar. Basierend auf aktuellen Programmierparadigmen, wie dem Model-View-Controller-Konzept<sup>44</sup>, wird bei der Entwicklung grundsätzlich auf eine modulare und damit erweiterbare Struktur gesetzt. Dies wird durch die objektorientierte Programmierung mittels PHP 5 ermöglicht.<sup>45</sup> Die Generierung der Webdateien ist aufwändig. Datensätze werden aus der Datenbank geladen, Grafiken werden erstellt und Konfigurationsdateien werden ausgelesen, um nur ein paar rechenintensive Mechanismen aufzuzählen.<sup>46</sup> Damit diese rechenintensiven Prozesse beschleunigt werden, bringt TYPO3 zahlreiche Funktionalitäten mit. Werden diese, zum Teil versteckten, Funktionen bewusst eingesetzt, können auch bei dieser Web-Applikation Leistungsreserven freigeschaltet werden. Aufgrund der vielen Erweiterungen, die für TYPO3 über das Online Repository<sup>47</sup> verfügbar sind, soll nur auf eine Standard-Installation eingegangen werden.

---

<sup>42</sup> Stand: August 2009

<sup>43</sup> Vgl. TYPO3 Association: „TYPO3 Download“, o.J. <http://typo3.org/download/packages/>, 04.08.2009

<sup>44</sup> Vgl. Schmidt 2007, S. 237

<sup>45</sup> Vgl. Kannengiesser 2007, S. 225 ff.

<sup>46</sup> Vgl. Ripfel/Meyer/Höppner 2008, S. 193

<sup>47</sup> Siehe: TYPO3 Association: „Online Repository“: <http://typo3.org/extensions/repository/>, 22.08.2009

## 4 Apache

Wie in der Problemstellung bereits angedeutet, ist der Apache HTTP Server ein Gemeinschaftsprodukt vieler freier Entwickler, die ihre Arbeit über Mailinglisten koordinieren. Damit die Arbeit mit dem Apache beginnen kann, ist zunächst eine Installation des Softwarepaketes nötig, die nun kurz angedeutet wird.

Der Webserver steht auf einer Unterseite der offiziellen Internetpräsenz, der Apache Software Foundation kostenlos zum Download bereit.<sup>48</sup> Neben der aktuellen Version 2.2.13<sup>49</sup>, die für den Produktiveinsatz von TYPO3 genutzt werden sollte, existiert eine veraltete, aber dennoch stabile Variante der ersten Generation.<sup>50</sup> Wer aufgrund seiner Server-Konfiguration keine Einschränkungen hinnehmen muss, sollte auf die neuste stabile Version von Apache zurückgreifen. Die meisten Linux-Distributionen bieten in ihrer Paketverwaltung die Möglichkeit, kompilierte Versionen des Web-servers, auf einfache Art und Weise, zu installieren. Das in dieser Arbeit favorisierte Debian 5.0, was auch unter dem Namen Lenny bekannt ist, stellt den HTTP-Server in Version 2.2.9 über seinen Paketmanager `apt`<sup>51</sup> zur Verfügung.<sup>52</sup>

Auf Windows-Systemen kann der Apache grundsätzlich auch eingesetzt werden. Ein entsprechendes Archiv steht ebenfalls zum Download bereit. Microsoft bietet allerdings auch eine eigene Web-Server-Lösung an. Die auf den Namen Internet Information Services getauft Lösung, wird in vielen von Microsoft entwickelten Betriebssystemen in eingeschränktem Maße mitgeliefert.<sup>53</sup>

Darüberhinaus sind viele weitere Webserver im Internet erhältlich, die mit geringem Ressourcenhunger werben. In diesem Zuge sei der `lighttpd`-Webserver erwähnt. Für den schlanken und zudem freien Webserver sind zahlreichen Anleitungen zur Installation im Internet verfügbar.<sup>54</sup>

### 4.1 Diagnose

Zu Beginn aller Optimierungsmaßnahmen steht die Diagnose. Bei der großen Anzahl der Komponenten auf einem Server ist es wichtig zu wissen, welche Software für den langsamen Seitenaufbau beim User sorgt. Für die Ermittlung eines solchen Flaschenhalses bringt der Apache-Webserver nützliche Diagnoseprogramme mit.

<sup>48</sup> Vgl. Apache Software Foundation: „Apache HTTP Server Project“, o.J., <http://httpd.apache.org/>, 12.08.2009

<sup>49</sup> Stand: August 2009

<sup>50</sup> Vgl. Apache Software Foundation: „Downloading the Apache HTTP Server“, <http://httpd.apache.org/download.cgi>, 05.08.2009

<sup>51</sup> Siehe Silva: „APT HOWTO“, 04/2003, <http://www.debian.org/doc/manuals/apt-howto/index.de.html>, 19.08.2009

<sup>52</sup> Vgl. Debian Projekt: „Paket: apache2 (2.2.9-10+lenny4)“, o.J., <http://packages.debian.org/lenny/apache2>, 05.08.2009

<sup>53</sup> Vgl. Microsoft: „Internet Information Services“, o.J., <http://www.iis.net/>, 05.08.2009

<sup>54</sup> Vgl. o.V.: „Webserver Lighttpd unter Debian 5.0 „Lenny“ Howto“, 17.06.2009, <http://debian.asconix.com/lighttpd-webserver-debian-lenny-howto>, 06.08.2009

Sie liefern neben Werten zur Messung der Performance, auch eine übersichtliche Darstellung der Einstellungen in der Konfigurationsdatei, die momentan verwendet wird. Dies ist nur ein grober Abriss der Funktionen, die diese kleinen Helfer leisten. Im Laufe dieses Abschnitts werden die Hilfsprogramme näher erläutert.

#### 4.1.1 Das Diagnose-Tool „ab“

Das erste nützliche Diagnose-Tool heißt `ab`. Das Programm wird über die Kommandozeile unter Linux, also über SSH bedient und ermöglicht es Performance-Tests mit einer beliebigen URL durchzuführen. Das im Unterordner `bin` zu findende Hilfsprogramm erlaubt so beispielsweise einen Aufruf von 1.000 Anfragen, die in 10 parallellaufenden Prozessen an `www.example.com` erfolgen.<sup>55</sup> Der dafür nötige Befehl sieht so aus:

```
ab -n 1000 -c 10 http://www.example.com
```

`ab` gibt in seinem Ergebnis neben obligatorischen Werten, wie Server-Software, Hostname und Port auch noch folgende Daten zurück:

```
Document Path:      /
Document Length:    16289 bytes
Concurrency Level:   1
Time taken for tests: 16.885975 seconds
Complete requests:   10
Failed requests:     0
Write errors:        0
Total transferred:   166570 bytes
HTML transferred:    162890 bytes
Requests per second: 0.59 [#/sec] (mean)
Time per request:    1688.597 [ms] (mean)
Time per request:    1688.597 [ms] (mean, across all
                    concurrent requests)
Transfer rate:       9.59 [Kbytes/sec] received
Connection Times (ms)
                min  mean[+/-sd] median  max
Connect:        353  375  16.1    386   391
Processing:    1240 1312  52.1   1339  1369
Waiting:        449  472  16.2    476   499
Total:         1593 1687  67.7   1730  1756

Percentage of the requests served within a certain time
(ms)

50%    1730
66%    1733
75%    1741
80%    1753
```

<sup>55</sup> Kersken, 2009, S. 532 ff.

90%	1756
95%	1756
98%	1756
99%	1756
100%	1756 (longest request) <sup>56</sup>

Das Hilfsprogramm `ab` liefert, wie man dieser Auflistung entnehmen kann, zahlreiche Ergebnisse zur Beurteilung der Leistung, des Servers, und auch der Bandbreite. Damit lassen sich Änderungen und vorgenommenen Maßnahmen zur Optimierung durch harte Fakten, i.e. Zahlen bestätigen oder widerlegen.

#### 4.1.2 Das Diagnose-Modul „mod\_info“

Neben dem eigenständigen Programm `ab` stellt der Apache 2, zwei Module für die Diagnose des Servers zur Verfügung. Eins davon ist die Erweiterung `mod_info`. Sie gibt die Konfigurationsdateien des momentan laufenden Apache-HTTP-Servers als HTML-Dokument über den Browser aus. Die Suche durch unzählige Konfigurationsdateien wird damit erleichtert. Darüberhinaus kann man sich eine Liste der derzeit aktivierten Module ausgeben lassen, was im Abschnitt **4.3. Einbindung von Modulen** noch relevant sein wird. Dort geht es darum nur die wirklich benötigten Module zu aktivieren.<sup>57</sup>

Das Modul ist auf einem Debian-System für gewöhnlich deaktiviert. Über die Kommandozeile lässt es sich mit dem Befehl `a2enmod info` aktivieren. Im Anschluss muss die Erweiterung für Zugriffe von außen verfügbar gemacht werden. Dafür ist folgender Eintrag in der Konfigurationsdatei von Nöten:

```
<Location /info>
    SetHandler server-info
</Location>
```

Wichtig dabei ist, dass der Pfad, in diesem Fall „/info“ keinem existierenden Verzeichnis entspricht. Durch den Eintrag in der Konfigurationsdatei wird nämlich eine Verknüpfung zur Erweiterung hergestellt. Die Server-Informationen sind nach einem Neustart des Webserver, über `http://www.example.com/info`, im Browser erreichbar.<sup>58</sup> Um den Zugriff vor Fremden zu schützen, kann innerhalb der `Location`-Tags noch eine Benutzer-Authentifizierung aktiviert werden<sup>59</sup>.

<sup>56</sup> Vgl. nixCraft: „Howto: Performance Benchmarks a Webserver“, o.J., <http://www.cyberciti.biz/tips/howto-performance-benchmarks-a-web-server.html>, 07.08.2009

<sup>57</sup> Kersken 2009, S. 362 f.

<sup>58</sup> Vgl. Apache Software Foundation: „Apache Module mod\_info“, o.J. [http://httpd.apache.org/docs/2.2/mod/mod\\_info.html](http://httpd.apache.org/docs/2.2/mod/mod_info.html), 31.07.2009

<sup>59</sup> Vgl. Apache Software Foundation: „Authentication, Authorization and Access Control“, o.J., <http://httpd.apache.org/docs/2.2/howto/auth.html>, 31.07.2009

### 4.1.3 Das Diagnose-Modul „mod\_status“

Das zweite interessante Diagnose-Modul ist `mod_status`. Es liefert nach Aufruf über den Browser interessante Informationen über aktuell laufende Prozesse bzw. Threads des Apache-http-Servers. Mit der Direktive `ExtendedStatus On` erhalten Sie zudem zusätzliche Informationen zu jeder einzelnen Anfrage zum Server. Zu beachten ist hierbei, dass diese Direktive zusätzliche Performance kostet. Sie sollte also nur aktiviert werden, wenn sie für die Arbeit wirklich benötigt wird.<sup>60</sup>

Die Aktivierung des Moduls läuft analog zu `mod_info` ab. Nach der Aktivierung mit `a2enmod status` über SSH müssen folgende Zeilen in die Konfigurationsdatei eingetragen werden:

```
<Location /status>
    SetHandler server-status
</Location>
```

Durch diese Konfiguration sind die Statusinformationen über die Adresse `http://www.example.com/status` abrufbar. Hier sollte ebenfalls über einen Zugriffsschutz nachgedacht werden, um die sensiblen Daten vor dem Einblick Dritter zu schützen.

## 4.2 Multi-Processing-Module

In der Apache-Version 2.0 haben die Enzwickler das MPM-Konzept erstmals eingeführt. Dieses Konzept dient dazu die Leistung bei den unterschiedlichen Betriebssystemen, auf denen der Apache-Webserver betrieben werden kann, zu verbessern. So stehen beispielsweise spezielle MPMs zur Verfügung, die auf die Architektur des jeweiligen Betriebssystems zugeschnitten sind. Für die Geschichte des Webserver bedeutete dies, dass der allgemein programmierte Kern des Webserver, der die Funktionalität für alle Betriebssysteme bereithielt, somit einem spezialisiertem MPM weichen musste.<sup>61</sup> Auf einem Debian-System hat man gegenwärtig die Wahl zwischen zwei verschiedenen MPMs, die ihre Aufgabe auf unterschiedliche Art und Weise lösen.<sup>62</sup>

### 4.2.1 prefork

Das klassische Preforking-Konzept erzeugt beim Start eine vordefinierte Anzahl von Prozessen, die für die Verarbeitung von Anfragen zuständig sind. Jeder dieser

<sup>60</sup> Vgl. Kersken 2009, S. 362 ff.

<sup>61</sup> Vgl. Kersken 2009, S. 127 ff. und Vgl. Apache Software Foundation: „Multi-Processing-Module (MPMs)“, o.J., <http://httpd.apache.org/docs/2.2/mpm.html>, 12.08.2009

<sup>62</sup> Vgl. Debian Projekt: „Paket: apache2 (2.2.9-10+lenny4)“, o.J., <http://packages.debian.org/lenny/apache2>, 12.08.2009 und Vgl. Apache Software Foundation: „Apache-MPM prefork“, o.J. <http://httpd.apache.org/docs/2.2/mod/prefork.html>, 12.08.2009

Prozesse, die auch *Forks* genannt werden, nutzt dabei seinen eigenen Speicherbereich. Dies wird später in Kapitel 5.2.1 **mod\_php** zum Tragen kommen, wenn es darum geht eine passende PHP-Integration zu finden. Die Preforking-Prozesse laufen parallel und verarbeiten dabei jeweils eine Verbindung. Somit ist die Anzahl der zu verarbeitenden Verbindungen zunächst begrenzt. Das Konzept ermöglicht allerdings die Angabe einer Mindestanzahl von freien Prozessen. Wird diese Anzahl unterschritten, werden automatisch neue Prozesse erzeugt. Da jeder Prozess seine eigenen Variablen und Dateideskriptoren benötigt, ist diese Erzeugung sehr aufwändig. Das Preforking-Konzept gilt als das klassische Modell, mit dem der Apache vor der Version 2.0 ausschließlich gearbeitet hat.<sup>63</sup> Dieses Konzept wird bei der Installation des Debian-Pakets `apache2` automatisch installiert.

#### 4.2.2 worker

Das zweite, zur Verfügung stehende, Modul baut ebenfalls auf parallel arbeitenden Prozessen auf. Innerhalb dieser Prozesse sorgen Threads für die Verarbeitung von Anfragen. Threads haben den Vorteil, dass sie im selben Speicherbereich arbeiten und somit auf die gleichen Variablen zugreifen. Dadurch können zusätzlich benötigte Threads schneller erzeugt werden, als Prozesse.<sup>64</sup> Der Nachteil dieses Konzepts, besteht darin, dass die Threads aufgrund gemeinsam genutzter Ressourcen nicht untereinander abgeschirmt sind.<sup>65</sup> Bei schlechter Programmierung kann dies zu Sicherheitsproblemen führen. Diese sind bei einer TYPO3-Installation eher zu vernachlässigen, solange man auf getestete Erweiterungen setzt. Im Paketmanager von Debian findet man dieses MPM unter dem Namen `apache2-mpm-worker`. „Es wird besonders empfohlen für Websites mit hohem Verkehrsaufkommen, da es schneller ist als das althergebrachte MPM »prefork« und weniger Speicher beansprucht“<sup>66</sup>, heißt es auf der Debian-Website. Eine ordnungsgemäße Konfiguration vorausgesetzt, arbeitet der worker also ressourcenschonender, als der prefork.<sup>67</sup>

#### 4.2.3 Ausblick auf Apache 2.3/2.4

Die nächste Version des Apache-HTTP-Servers wird bereits entwickelt. Sie wird im Alpha- oder Beta-Stadium noch die Nummer 2.3 tragen, bevor sie als stabil gilt und in Version 2.4 umbenannt wird. Aus Sicht der Entwickler hat sich das Konzept der MPM

<sup>63</sup> Vgl. Kersken, 2009, S. 128

<sup>64</sup> Vgl. Apache Software Foundation: „Apache-MPM worker“, o.J., <http://httpd.apache.org/docs/2.2/mod/worker.html>, 12.08.2009

<sup>65</sup> Vgl. o.V.: „Apache2/workerMPM/FastCGI/PHP5“, 04.09.2008, <http://www.digitalnerds.net/featured/apache2-worker-mpm-with-fastcgi-php5/>, 12.08.2009

<sup>66</sup> Debian Projekt: „Paket: apache2-mpm-worker (2.2.9-10+lenny4)“, o.J., <http://packages.debian.org/lenny/apache2-mpm-worker>, 14.08.2009

<sup>67</sup> Vgl. Debian Projekt: „Paket: apache2-mpm-worker (2.2.9-10+lenny4)“, o.J., <http://phpperformance.de/prozesskonfiguration/>, 22.07.2009

noch nicht durchgesetzt. Der Großteil der Server setzt immer noch auf das Preforking-Konzept. Daher haben die Apache-Entwickler `mpm_simple` entwickelt, welches automatisch das jeweils schnellste MPM auswählt.<sup>68</sup>

Im Vorangegangenen wurden die zwei wichtigsten *Multi-Processing-Modules* und ihre Funktionsweise beschrieben. Damit sollte deutlich geworden sein, welche Variante für den Einsatz in der eigenen Serverkonfiguration benötigt wird.

### 4.3 Einbindung von Modulen

Der Apache-Webserver bringt durch seine zahlreichen Module vielfältige Funktionalität mit sich. Für die Verwendung der Module gibt es zwei verschiedene Arten der Integration. Auf der einen Seite gibt es die Möglichkeit die benötigten Module bei der Installation direkt mit-zu-kompilieren. Hier spricht man vom statischen Einbinden, da die Module nach der Installation direkt in den Webserver integriert sind. Sollte sich das Anforderungsprofil des Servers durch die Website ändern, wäre eine Neuinstallation des Webserver von Nöten. Zu diesem Zweck haben die Entwickler das *Dynamic Shared Object* entwickelt. Durch diese Moduleinbindung ist der Apache-HTTP-Server in der Lage, die erforderlichen Module nach Bedarf zu integrieren. Dies geschieht über die Konfigurationsdatei des Apachen, die dafür die `LoadModule`-Direktive zur Verfügung stellt. Auf einem Debian-System wird die Integration mittels Konsole gelöst. Der Befehl `a2enmod` aktiviert ein Modul, während `a2dismod` ein Modul wieder deaktiviert. Ein Neustart des Webserver ist in beiden Fällen von Nöten.<sup>69</sup>

Der Vorteil bei der statischen Integration ist die Geschwindigkeit. Auf der offiziellen Internetseite spricht man von Performance-Schüben von bis zu 5% bei der Ausführung eines Modul-Prozesses und eine bis zu 20% kürze Ladezeit des Moduls beim Starten des Webserver.<sup>70</sup> Dies ist auf die schlanke Architektur eines statisch integrierten Moduls, im Gegensatz zu einem dynamischen eingebundenen Moduls, zurückzuführen. Hier gilt es abzuwägen. Zudem ist eine heterogene Einbindung der Module möglich. Durch eine solche Einbindung lässt sich einerseits das für „sprechende URLs“<sup>71</sup> von TYPO3 benötigte Modul `mod_rewrite` beim Kompiliervorgang und zugleich auch etwaige andere Module über die dynamische Schnittstelle für DSOs integrieren. Aufgrund der strategischen Ausrichtung von Debian, nur wirklich benötigte Pakete zu installieren, zeigt sich der Apache in seiner Grundkonfiguration sehr aufgeräumt. Ein Eingreifen ist auf einem frisch installierten Debian-System also nicht nötig.

<sup>68</sup> Vgl. Kersken, 2009, S. 835 f.

<sup>69</sup> Vgl. Kersken, 2009, S. 531

<sup>70</sup> Vgl. Apache Software Foundation: „Dynamic Shared Object (DSO) Support“, o.J., <http://httpd.apache.org/docs/2.0/dso.html>, 22.07.2009

<sup>71</sup> Vgl. „Clean URLs“, [http://de.wikipedia.org/wiki/Clean\\_URLs](http://de.wikipedia.org/wiki/Clean_URLs), 23.07.2009



## 4.4 Komprimierung mit mod\_deflate

Einen anderen Ansatz zur Optimierung bietet das Apache-Modul `mod_deflate`. Es komprimiert die übermittelten Daten mithilfe des patentfreien, offenen Komprimierungsverfahrens GZip.<sup>72</sup> Durch diese Kompression reduziert sich die Größe der Server-Antwort um rund 70 %.<sup>73</sup> Allerdings geschieht dies zulasten der Serverressourcen. Das Modul arbeitet dabei als Filter und kann sowohl für den ein-, als auch ausgehenden Datenverkehr genutzt werden. Die ankommenden Daten werden dabei vom Input-Filter verarbeitet, wohingegen der Output-Filter sich um die ausgehenden Daten kümmert. Wichtig für diese komprimierte Kommunikation ist, dass beide Seiten (Sender/Empfänger) das GZip-Verfahren verstehen.<sup>74</sup> Aus diesem Grund liefert der Browser in seiner Anfrage einen entsprechenden *Accept-Header*, um dem Server zu vermitteln, dass er die komprimierte Kommunikation versteht. Der Server sendet daraufhin den komprimierten HTTP-Body mit dem angeforderten Inhalt und einem Hinweis auf GZip-Kompression. Alle modernen Browser verstehen dieser Art der Kommunikation. Nur die im Folgenden genannten Browser und ihre Vorversionen unterstützen GZip nur eingeschränkt oder gar nicht:

- Netscape 7
- Mozilla 1
- Internet Explorer 4
- Opera 5
- Lynx 2.6

Diese Browser gehören ganz klar zur aussterbenden Generation. Ihr Marktanteil ist so gering, dass er bewusst vernachlässigt werden kann. Wer trotzdem auf Nummer sicher gehen will, kann nach der Aktivierung von `mod_deflate`, in der Konfigurationsdatei vom Apache-Webserver Einschränkungen vornehmen. Durch diese Einstellungen werden nur unkomprimierte Inhalte an die Browser ohne GZip-Unterstützung gesendet.<sup>75</sup>

Durch den Einsatz von `mod_deflate` wird der Traffic einer TYPO3-Installation minimiert. Man sollte sich allerdings vor Augen führen, dass dadurch der Prozessor eine Mehrbelastung erfährt. Wenn der Server neben den für TYPO3 nötigen Komponenten noch genug Ressourcen für die Komprimierung besitzt und es an Bandbreite fehlt, kann dies allerdings die Ladezeiten der Website reduzieren.<sup>76</sup>

<sup>72</sup> Vgl. Kersken, 2009, S. 704 f. und

Vgl. Apache Software Foundation: „Apache Module mod\_deflate“, o.J., [http://httpd.apache.org/docs/2.2/mod/mod\\_deflate.html](http://httpd.apache.org/docs/2.2/mod/mod_deflate.html), 23.07.2009

<sup>73</sup> Vgl. Yahoo Developer Network: „Gzip Components“, o.J., <http://developer.yahoo.com/performance/rules.html#gzip>, 23.07.2009

<sup>74</sup> Vgl. Bold: „Ressourcen schonen“, 04/2008, 116 f.

<sup>75</sup> Apache Software Foundation: „Sample Configurations“, o.J., [http://httpd.apache.org/docs/2.2/mod/mod\\_deflate.html#recommended](http://httpd.apache.org/docs/2.2/mod/mod_deflate.html#recommended), 24.07.2009

<sup>76</sup> Vgl. Kersken, 2009, S. 705

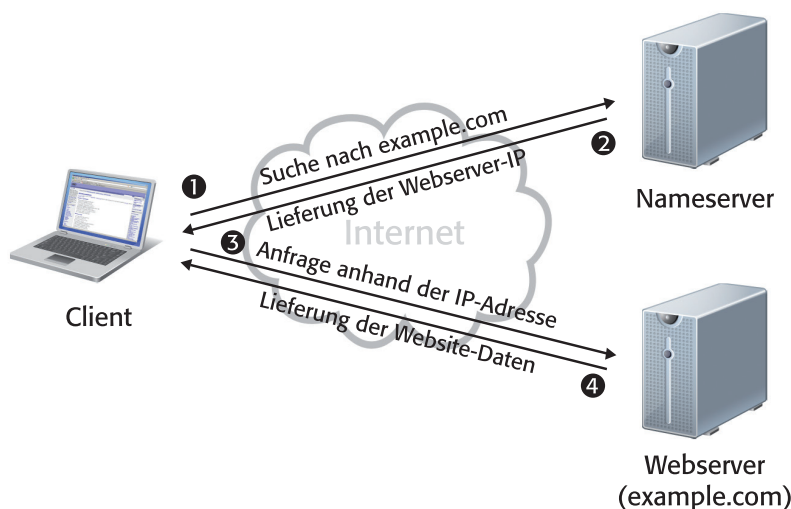


In Debian ist das Modul bei Installation des Apache-Pakets automatisch integriert. Es muss mit dem Befehl `a2enmod deflate` über die Konsole und in der `htaccess`-Konfigurationsdatei aktiviert werden.<sup>77</sup>

## 4.5 DNS-Lookups vermeiden

Hinter einem DNS-Lookup versteht man das Übersetzen einer Domain in eine IP-Adresse. Dieser wird beispielsweise beim Aufruf einer Website nötig: Um eine Verbindung zwischen Browser (Client-Seite) und Website (Server-Seite) aufzubauen zu können, kommuniziert der Browser zunächst mit einem Nameserver. Dieser liefert für eine Domain, die IP-Adresse des Servers, auf dem sich die Website befindet. Nun kann der Browser direkt mit dem Server, der Website kommunizieren und die Website-Daten anfordern.<sup>78</sup> Mit der folgenden Grafik soll das verdeutlicht werden.

Abbildung 1: Client-Server-Kommunikation<sup>79</sup>



Ähnlich funktioniert es auch auf umgekehrtem Wege. Beispielsweise lassen sich die IP-Adressen der User für Log-Dateien in Hostnamen umwandeln. Man spricht in diesem Fall von einem Reverse-DNS-Lookup, da es genau entgegengesetzt zu dem oben Beschriebenen abläuft. Diese Arbeit kann der Apache-Webserver mit der Direktive `HostNameLookups` übernehmen. Allerdings kostet diese Befragung eines Nameservers Zeit. Mit der Einstellung `HostNameLookups Off` werden diese Nameserver-Anfragen unterbunden und lediglich die IP-Adresse in den Logdateien protokolliert.<sup>80</sup>

<sup>77</sup> Vgl. Timme: „Bandbreite sparen mit Apache2s mod\_deflate“, 15.05.2009, [http://www.howtoforge.de/howto/bandbreite-sparen-mit-apache2s-mod\\_deflate/](http://www.howtoforge.de/howto/bandbreite-sparen-mit-apache2s-mod_deflate/), 24.07.2009

<sup>78</sup> Vgl. Kersken, 2009, S. 32 ff.

<sup>79</sup> In Anlehnung an: <http://www.e-teaching.org/technik/vernetzung/architektur/client-server/webserver>, 19.08.2009

<sup>80</sup> Vgl. Kersken, 2009, S. 487

Bei der Anpassung dieser Einstellung muss abgewogen werden, ob die weniger detaillierten Zugriffstatistiken, die Einsparungen der Bandbreite rechtfertigen. TYPO3 benötigt diese Einstellung grundsätzlich nicht. Neben den Optionen `on` und `off` kann die Einstellung `Double`, für die doppelte Überprüfung, als übertrieben angesehen werden.<sup>81</sup>

---

<sup>81</sup> Vgl. Apache Software Foundation: „HostnameLookups-Direktive“, o.J., <http://httpd.apache.org/docs/2.0/mod/core.html#hostnamelookups>, 25.07.2009

## 5 PHP

Für die Verwendung von TYPO3 wird die Skriptsprache PHP benötigt. Anders als beim Webserver hat man hier nicht die Wahl zwischen verschiedenen Lösungen. Auch für die zukünftigen Versionen von TYPO3 ist zu erwarten, dass das System weiterhin auf die populäre Skriptsprache setzt. Daher ist es wichtig zu wissen, welche Verarbeitungsschritte ein aufgerufenes PHP-Skript durchläuft.

Die Programmiersprache PHP wurde im Jahr 1995 erstmals der Öffentlichkeit präsentiert. Damals noch bekannt unter dem Namen „Personal Home Page Tools“, wurden die ersten Versionen durch Rasmus Lerdorf entwickelt.<sup>82</sup> Ziel war es, mit PHP eine Möglichkeit für die schnelle und dynamische Erstellung von Webseiten zu liefern<sup>83</sup>. Die Erstellung von PHP-Dateien erfordert prinzipiell keine teure Entwicklungsumgebung und kann auch mit dem Texteditor vorgenommen werden.<sup>84</sup> Die Möglichkeit PHP in HTML-Code zu integrieren, kennzeichnet die Programmiersprache zudem als Metasprache. Der PHP-Interpreter ist in C geschrieben und übersetzt die geschriebenen Skripte bei jedem Aufruf erneut. Das PHP-Projekt wird von einer aktiven Community kontinuierlich weiterentwickelt. Die Entwickler selber, beschreiben die Skriptsprache, wie folgt:

„PHP ist Klebstoff. Es ist der Klebstoff, der verwendet wird, um coole Web-Anwendungen zu erstellen indem er dutzende von Bibliotheken von Drittanbietern zusammenklebt und sie über eine intuitiv und einfach erlernbare Schnittstelle als zusammenhängende Einheit erscheinen lässt“.<sup>85</sup>

Diese Einschätzung auf der offiziellen PHP-Website, verdeutlicht, warum so viele Web-Applikation als robuste und performante Basis, auf PHP setzen. So setzt auch TYPO3, seit der ersten öffentlichen Version, auf PHP. Anfangs auf PHP 4 aufbauend, benötigt das CMS aktuell mindestens die Version 5.2.<sup>86</sup> Dies resultierte aus einem Projekt zur Förderung von Applikationen, die auf Programmiersprache PHP 5 basieren.<sup>87</sup> Mit der aktuellen, stabilen Generation der Skriptsprache basiert TYPO3 auf einer modernen und objektorientierten Architektur.

Zugleich steigt auch die Komplexität der Skripte durch das in der Einleitung thematisierte Wachstum der Funktionalitäten bei Webanwendungen. Um mit dieser Komplexität umgehen zu können, sind zahlreiche Lösungsansätze in der Informatik entworfen worden. Diese schreiben vor, wie man Anwendungen in seine Teilbereiche zerlegt, um die Übersicht und die

<sup>82</sup> Vgl. Haas/Kücükyilmaz/Merz 2005, S. 8

<sup>83</sup> Vgl. The PHP Group: „Vorwort“, 07.08.2009, <http://de3.php.net/manual/de/preface.php>, 25.07.2009

<sup>84</sup> Vgl. Letzel/Gacki 2001, S. 43

<sup>85</sup> Vgl. The PHP Group: „Installation“, 07.08.2009, <http://de3.php.net/manual/de/faq.installation.php#faq.installation.apache2>, 26.07.2009

<sup>86</sup> Vgl. Stucki: „Leaving PHP4 behind“, 13.07.2007, <http://buzz.typo3.org/people/stucki/article/leaving-php4-behind/>, 12.08.2009

<sup>87</sup> Vgl. GoPHP5.org: „Home“, 05.02.2008 <http://gophp5.org/>, 24.07.2009

Erweiterbarkeit nicht zu verlieren. Die oft als kompliziert bezeichneten Programmierparadigmen<sup>88</sup> finden natürlich auch bei so einem vielfältigen Content Management System, wie TYPO3 ihre Verwendung. Vielerorts hört man von Entwurfsmustern, wie dem Model-View-Controller-Konzept, das die strikte Trennung der Geschäftslogik, von der Präsentationsschicht und der Steuereinheit beschreibt.<sup>89</sup> Bei all diesen Grundsätzen darf man nicht vergessen, dass jede weitere Abstraktionsebene Ressourcen frisst. Es gilt somit, neben dem bereits beschriebenen Webserver, nun auch den „Klebstoff“<sup>90</sup> PHP zu beschleunigen. Die Ansätze der Optimierung sind mannigfaltig und werden in folgenden Unterkapiteln genauer erörtert.

Wichtig zu erwähnen ist an dieser Stelle, dass nicht in den Quellcode von TYPO3 eingegriffen wird und dies auch nicht zu empfehlen ist. Zum einen wären sehr gute Kenntnisse von PHP und TYPO3 von Nöten und zum anderen würde ein Update des CMS diese Anpassungen wieder zunichte machen.

## 5.1 PHP-Version

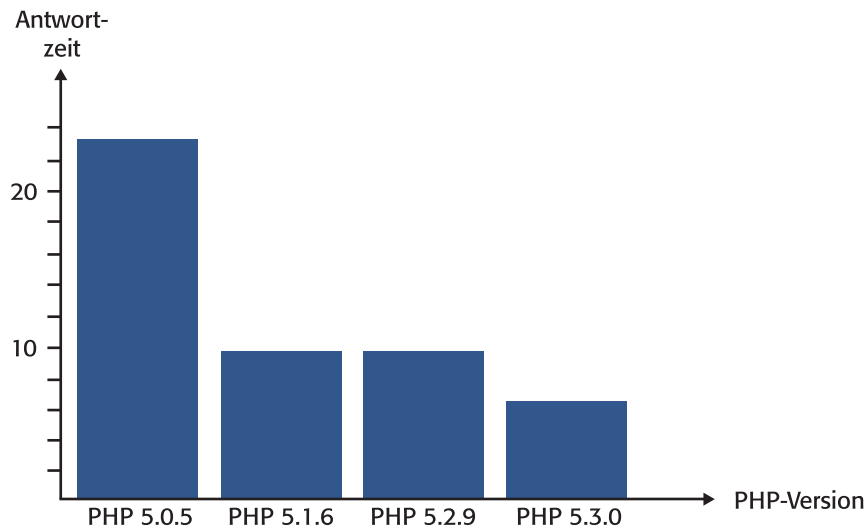
Wie bereits erwähnt, erfordert das aktuelle TYPO3 die PHP-Version 5.2 oder höher. Wer auf eine ältere Version des Content Management Systems setzt, sollte sich über die erforderliche Mindestversion informieren. Wie bei den meisten Software-Paketen bringt eine aktuelle Version, neben neuen Funktionen und Bugfixes, auch Performance-Verbesserungen mit sich. Dies ist auf das Refactoring<sup>91</sup> zurückzuführen, welches zum guten Ton einer jeden Software-Entwicklung dazugehört. Vereinfacht dargestellt, werden komplexe Algorithmen durch simplere ersetzt, die jedoch das Gleiche tun. Bei den verschiedenen PHP-Versionen lässt sich dieser Umstand, durch folgende Grafik verdeutlichen.

<sup>88</sup> Kannengiesser 2007, S. 373 ff.

<sup>89</sup> Schmidt, 2007, S. 233 ff.

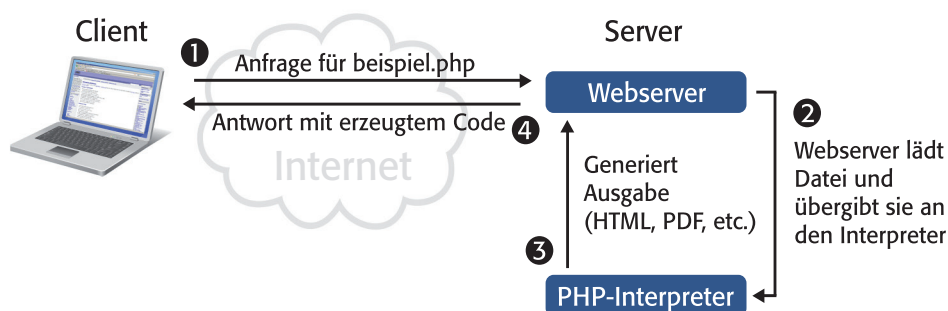
<sup>90</sup> The PHP Group: „Installation“, 21.07.2009, <http://de3.php.net/manual/de/faq.installation.php#faq.installation.apache2>, 26.07.2009

<sup>91</sup> Siehe: „Refactoring“, <http://de.wikipedia.org/wiki/Refactoring>, 26.07.2009

Abbildung 2: PHP5 im Vergleich<sup>92</sup>

## 5.2 Einbindung des PHP-Interpreters

Um die nun folgenden Maßnahmen zur Optimierung der Performance, zu verstehen, ist es notwendig zu wissen, wie die Verarbeitung eines PHP-Skripts von Statten geht. Es beginnt mit der Anfrage durch den User. Diese gelangt zum Server, auf dem das PHP-Skript liegt. Der Webserver öffnet daraufhin die PHP-Datei und leitet sie an den PHP-Interpreter weiter. Von ihm wird das PHP-Skript abgearbeitet. Das Resultat aus dieser Interpretation wird dem Webserver zurückgegeben. Daraufhin gelangt das Ergebnis zurück zum User. Der Prozess wird nachfolgend durch eine Grafik exemplarisch dargestellt.

Abbildung 3: Funktionsweise von PHP<sup>93</sup>

<sup>92</sup> In Anlehnung an heise Developer: „Benchmark of PHP 5 Versions“, 30.06.2009, <http://www.heise.de/developer/Was-aendert-sich-mit-PHP-5-3-/zoom/140003/2>, 19.08.2009

<sup>93</sup> In Anlehnung an: „PHP Funktionsweise“, 01.07.2009, [http://de.wikipedia.org/wiki/Datei:PHP\\_funktionsweise.svg](http://de.wikipedia.org/wiki/Datei:PHP_funktionsweise.svg), 19.08.2009

Für die Kommunikation zwischen PHP-Interpreter und Webserver gibt es verschiedene Arten der Implementierungen. Jede dieser Einbindungen hat ihre Vor- und Nachteile, die im Verlauf der folgenden Unterkapitel näher erörtert werden sollen.<sup>94</sup>

### 5.2.1 mod\_php

Die erste Möglichkeit PHP in die Verarbeitung der Webdateien zu integrieren, ist die Verwendung des Apache-Moduls *mod\_php*. Wie bereits in **Kapitel 4.1.2 mod\_info** beschrieben, gestaltet sich die Aktivierung eines solchen DSOs sehr einfach. Die Einbindung von PHP via *mod\_php* gehört daher bei vielen Linux-Distributionen, wie auch Debian zur Standardimplementierung. Die 5. Generation der Skriptsprache findet man im Paketmanager von Debian unter dem Namen `php5`.<sup>95</sup> Neben der schnellen Installation, bietet *mod\_php* eine schnelle Interpretation der PHP-Skripte. Diese schnelle Bearbeitung fällt allerdings zulasten der Sicherheit. PHP-Skripte werden unter Linux mit den Rechten des Webserver ausgeführt. Dadurch ist es dem PHP-Interpreter möglich, auf alle Daten mit den Rechten des Webserver zuzugreifen. Dies würde beim Betreiben eines Servers mit mehreren Kunden für ein erhebliches Sicherheitsrisiko sorgen, da die Kunden gegenseitig auf ihre Dateien zugreifen könnten. Dazu kommt, dass diese Integration als Apache-Modul, bei einem Fehler im Interpreter, den ganzen Apache zum Absturz bringen kann.

Des Weiteren setzt *mod\_php* die Installation des Apache-Webserver als Preforking-Server voraus, wie im Kapitel **4.2.1 prefork** bereits beschrieben. Dies hat den Nachteil, dass jeder erzeugte Fork des Apaches, den benötigten PHP-Interpreter im Speicher hat. Die Auslastung des Arbeitsspeichers ist dadurch enorm. Die schnelle Verarbeitung der Skripte wird bei *mod\_php* also durch einen hohen Speicherverbrauch und mangelnde Sicherheit erkaufte. Daher ist die Nutzung dieser Variante nur bedingt anzuraten. Wer keinerlei Speicherprobleme bei der Auslieferung seiner Website hat und zudem einen alleinigen Server für seine Website nutzt, kann *mod\_php* nutzen. Allerdings erfordert ein späterer Umbau dieser Konfiguration eine Neuinstallation von PHP und dem Apache-HTTP-Server.

### 5.2.2 PHP als CGI via suPHP

Aus der beschriebenen Rechteproblematik von *mod\_php* heraus, entstand die Verwendung von PHP als CGI-Modul. Das *Common Gateway Interface* dient als standardisierte Schnittstelle für eine Vielzahl von Programmiersprachen, wie Perl und Python. Für die Nutzung dieser Variante ist zudem noch das Modul *suPHP*<sup>96</sup> nötig. Es regelt

<sup>94</sup> Vgl. o.V.: „mod\_php vs. PHP-CGI“, 08.07.2009 [http://wiki.rootforum.de/scripting/php/mod\\_php\\_vs\\_php-cgi](http://wiki.rootforum.de/scripting/php/mod_php_vs_php-cgi), 27.07.2009

<sup>95</sup> Vgl. Debian Projekt: „Paket: php5 (5.2.6.dfsg.1-1+lenny3)“, o.J., <http://packages.debian.org/lenny/php5>, 27.07.2009

<sup>96</sup> Siehe: Marsching: „suPHP“, 14.03.2009, <http://www.suphp.org/Home.html>, 28.07.2009

die Ausführung der PHP-Skripte mit den Rechten des Besitzers. Dadurch müssen die Skripte nicht, wie bei *mod\_php*, mit den Rechten des Webserver ausgeführt werden, was einem enormen Sicherheitszuwachs entspricht.<sup>97</sup> Einem Server mit mehreren Kunden-Accounts steht somit nichts mehr im Wege. Zu erwähnen wäre noch, dass PHP via CGI, eine Verwendung von verschiedenen PHP-Versionen ermöglicht. Für die Verwendung von TYPO3 spielt das grundsätzlich keine Rolle. Sollten allerdings Internetpräsenzen auf Basis des überholten PHP4 auf dem Server liegen, kann je nach Dateieindung der benötigte PHP-Interpreter genutzt werden.<sup>98</sup> Im Gegensatz zu *mod\_php* benötigt PHP als CGI die Verwendung des Apache-Workers, der bereits in Kapitel 4.2.2 **worker** näher erörtert wurde.<sup>99</sup>

Größtes Manko von PHP als CGI via suPHP ist die schlechte Performance. Diese bleibt weit hinter der von *mod\_php* zurück. Das liegt daran, dass mit jedem Seitenaufruf der PHP-Interpreter in den Prozess geladen und nach Abschluss der Interpretation wieder entladen werden muss. Dies sorgt für viele Zugriffe auf der langsamen Festplatte und den Verbrauch von Systemressourcen. Die PHP-Implementierung via suPHP ist daher für den Einsatz in einer TYPO3-Konfiguration, die auf viele Nutzer ausgelegt sein soll, nicht zu empfehlen. Die PHP-Entwickler raten in ihrer Dokumentation ebenfalls davon ab.<sup>100</sup>

Die Auflistung dieser Implementierung dient daher der Vollständigkeit. Wichtig ist, dass bei der Installation unter Debian explizit das Paket `php5-cgi` angegeben wird. Durch die Abhängigkeit in der Paketverwaltung wird automatisch das Paket für den Apache-Worker (`apache2-mpm-worker`) mit installiert.

### 5.2.3 PHP als CGI via mod\_fcgid

Bei der letzten Implementierung handelt es sich um eine Variante, die die Vorzüge von *mod\_php* und PHP via suPHP vereint. Das Modul *mod\_fcgid* setzt dabei auf die Verwendung der Schnittstelle FastCGI. Dadurch kommt die Performance von PHP via *mod\_fcgid* fast an die einer *mod\_php*-Integration heran. *mod\_fcgid* muss man sich als eine Art Schleife vorstellen, die auf zu interpretierende PHP-Skripte wartet.<sup>101</sup> Der PHP-Interpreter wird dabei permanent im Arbeitsspeicher gehalten und wird somit nicht bei jedem Seitenaufruf erneut geladen.<sup>102</sup> Daraus resultiert eine Leistung, die

<sup>97</sup> Vgl. o.V.: „mod\_php vs. PHP-CGI“, 08.07.2009, [http://wiki.rootforum.de/scripting/php/mod\\_php\\_vs\\_php-cgi](http://wiki.rootforum.de/scripting/php/mod_php_vs_php-cgi), 28.07.2009

<sup>98</sup> Vgl. o.V.: „Apache: php4 und php5 parallel“, <http://admirableadmin.de/32/apache-php4-und-php5-parallel>, 28.07.2009

<sup>99</sup> Vgl. Ryan: „Difference between PHP thread safe and non thread safe binaries“, 27.09.2007, [http://www.iis-aid.com/articles/my\\_word/difference\\_between\\_php\\_thread\\_safe\\_and\\_non\\_thread\\_safe\\_binaries](http://www.iis-aid.com/articles/my_word/difference_between_php_thread_safe_and_non_thread_safe_binaries), 28.07.2009

<sup>100</sup> Vgl. The PHP Group: „Installation“, 21.07.2009, <http://de3.php.net/manual/de/faq.installation.php#faq.installation.apache2>, 28.07.2009

<sup>101</sup> Vgl. Walther: „Fast-CGI – was steckt dahinter?“, 21.05.2007, <http://phpperformance.de/fast-cgi-was-steckt-dahinter/>, 01.08.2009

<sup>102</sup> Vgl. „FastCGI“, <http://de.wikipedia.org/wiki/FastCGI#Funktionsweise>, 29.07.2009

trotz der Verwendung von PHP als CGI, nicht mit dem Einsatz von *suPHP* zu vergleichen ist. Eine differenzierte Rechtvergabe, wie sie bei einem System mit mehreren VHosts benötigt wird, ist zudem möglich. Die Kehrseite der Medaille ist in diesem Fall, dass es keine „on-the-Fly“-Installation für eine solche Konfiguration gibt. Während *mod\_php* unter Debian, in wenigen Schritten installiert und konfiguriert ist, benötigt die Einrichtung von PHP via *mod\_fcgid* einiges an Mehrarbeit. Dieser Aufwand soll jedoch nicht vor dem Gebrauch dieser Konfiguration abschrecken. Durch ihre Flexibilität lassen sich zusätzliche Kunden schnell in die bestehende Server-Konfiguration integrieren.<sup>103</sup>

Diese Aufwändige Konfiguration macht sich aber aus den bereits genannten Gründen bezahlt und dient daher auch vielen Hosting-Providern als Basis für die PHP-Einbindung. Für die Installation bei Debian wird das Paket *libapache2-mod-fcgid* benötigt. Auch hier werden wieder die Abhängigkeiten mit der automatischen Installation des Apache-Workers gelöst. Die Einbindung von PHP über FastCGI, ist darüberhinaus auch bei anderen Webservern, wie z. B. *lighttpd* möglich.

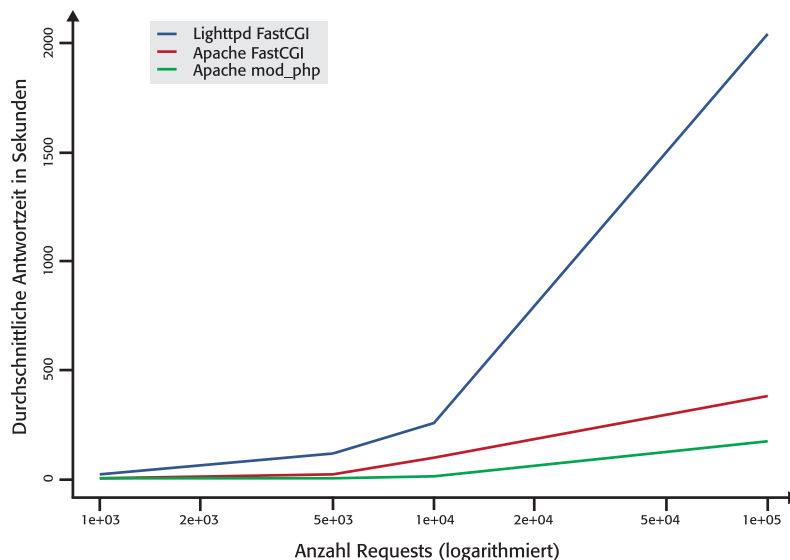
### 5.2.4 Vergleich

Abschließend soll mit der folgenden Grafik ein Vergleich ermöglicht werden, der die beiden relevanten Konfigurationen, *mod\_php* & PHP via *mod\_fcgid* gegenüberstellt. Damit wird die Behandlung dieses Themas abgeschlossen. Die Integration von FastCGI bei einem *Lighttpd*-Webserver dient hier zum Vergleich zu einem alternativen Webserver.<sup>104</sup> Der Graph zeigt hier die Abhängigkeit zwischen der Anzahl der Aufrufe (x-Achse) und der durchschnittlichen Antwortzeit in Sekunden (y-Achse). Für den Test wurde eine ca. 500 KB große PHP-Datei verwendet.

<sup>103</sup> Vgl. Stucki: „using PHP width *mod\_fcgid*“, <http://typo3.org/development/articles/using-php-with-mod-fcgid/page/2/>, 29.07.2009

<sup>104</sup> Vgl. nixCraft: „Lighttpd PHP fastcgi configuration“, o.J., <http://www.cyberciti.biz/tips/lighttpd-php-fastcgi-configuration.html> 30.07.2009

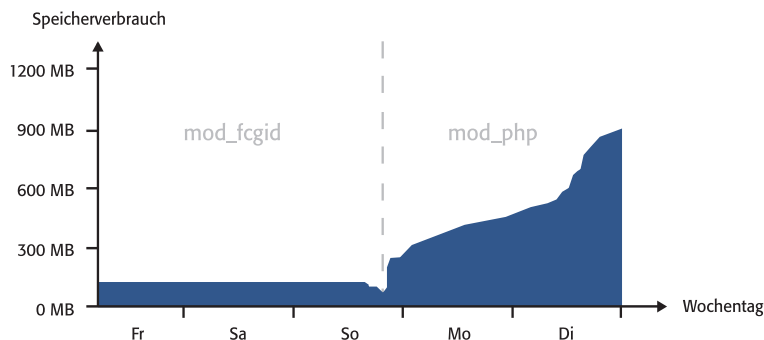


Abbildung 4: Auswertung dynamischen Contents mit 500 KB Dateien<sup>105</sup>

Wie man der Statistik entnehmen kann, liefert die Konfiguration mit `mod_php` die Website am schnellsten aus. Der Anstieg der Antwortzeit, ist auch bei großem Besucheraufkommen gering. Die FastCGI-Lösung ist bis zu einer Anfragenanzahl von ca. 5.000 Anfragen vergleichbar mit `mod_php`. Der Webserver `lighttpd` liegt aufgrund großer Belastung der Systemressourcen auf dem letzten Rang.

Diese Grafik unterstreicht die bereits geschilderte Leistung der Protagonisten `mod_php` und PHP als CGI via `mod_fcgid`. Zu beachten ist, dass die Grafik nicht zeigt, wie stark die Hardware beansprucht wird. Dies wird durch die folgende Statistik deutlich. Sie zeigt die Nutzung des Arbeitsspeichers im Verlauf der Zeit. Als Grundlage dient ein Apache-HTTP-Server. Der Graph zeigt von Freitag bis Sonntag die Speichernutzung einer `mod_fcgid`-Konfiguration. Montag bis Dienstag wird `mod_php` eingesetzt.

<sup>105</sup> In Anlehnung an Landschoof/Ostertag/Tyka: „Analyse von Systemperformanz“, o.J., <http://www2.net.in.tum.de/teaching/WS06/performprak/ausarbeitungen/Webserver-Auswertung.pdf>, 29.07.2009

Abbildung 5: Speicherverbrauch mod\_php vs. FastCGI<sup>106</sup>

Hier erkennt man, dass die Speicherauslastung bei der Verwendung von PHP via `mod_fcgid` konstant bleibt. Bei der Integration mit `mod_php` hingegen, steigt die Auslastung des Arbeitsspeichers stetig.

In diesem Abschnitt wurde somit gezeigt, dass `mod_fcgid` für die Integration einer hochfrequentierten Webseite besser geeignet ist. Die Auslieferungszeit ist nur unwesentlich langsamer als die von `mod_php`, aber der Speicherverbrauch ist um ein vielfaches geringer. Mit einer Konfiguration von PHP via FastCGI besitzt der Server mehr Leistungsreserven im Produktiveinsatz.

## 5.3 php.ini

Einstellungen am Verhalten des PHP-Interpreters werden in der Regel über eine zentrale Konfigurationsdatei erledigt. Bei der Verwendung von VHosts können diese auch separat für jede Website vorliegen. Auf einem Debian-System mit einer Internetpräsenz ist die Datei meist in `/etc/php4/cli/php.ini` zu finden. Für die Beschränkung der Einstellung auf einzelne Bereiche bietet sich zudem die Verwendung einer `htaccess`-Datei an, wenn das System das Setzen von PHP-Einstellungen auf diese Art erlaubt.

Darüberhinaus sei erwähnt, dass die in diesem Abschnitt erwähnten Anpassungen die Leistung nur minimal verbessern.

### 5.3.1 always\_populate\_raw\_post\_data

Hinter der Direktive `always_populate_raw_post_data` versteckt sich eine Möglichkeit an unbearbeitete Post-Daten zu kommen. Post-Daten werden zur Übermittlung von Formulareingaben benötigt. TYPO3 benötigt in seiner Standardinstallation

<sup>106</sup> In Anlehnung an o.V.: „Memory usage Apache + PHP as module versus FastCGI“, 23.05.2009, <http://www.apachelounge.com/viewtopic.php?p=10991>, 30.07.2009

aber keine Möglichkeit auf die unbearbeiteten Daten zuzugreifen.<sup>107</sup> Darüberhinaus stellt PHP andere Implementierungen bereit, um an diese Rohdaten zu gelangen.<sup>108</sup> Diese Funktion kann also ohne Bedenken deaktiviert werden.

### 5.3.2 max\_execution\_time

Diese Einstellung hat nur indirekt etwas mit der Leistung des Servers zu tun. `execute_time` bestimmt die Zeit in Sekunden, die ein PHP-Skript ausgeführt werden darf. TYPO3 verwendet viele komplexe Mechanismen, wie das Zwischenspeichern von Seiten oder Auslesen von Datensätzen aus der Datenbank. Um diese Aufgaben abzuarbeiten, benötigt der PHP-Interpreter Zeit. Ist die Einstellung für die maximal zur Verfügung stehende Zeit zu klein, bricht die Ausführung ab. Diese Option ist eigentlich für schlecht programmierte Skripte gedacht, die eventuell eine Endlosschleife produzieren. Ein optimaler Wert, um bei einer TYPO3-Installation, ein Skript komplett abzuarbeiten und dabei nicht unnötig lang die Systemressourcen zu strapazieren, ist 60 Sekunden.<sup>109</sup>

Im Erweiterungsmanager von TYPO3 können Extensions von externen Servern geladen werden um die Funktionalität vom CMS zu erweitern. Dies kann bei einer langsamen Verbindung und einer zu niedrig gewählten Ausführungszeit, zu Abbrüchen in der Skriptverarbeitung kommen, die mit folgender Fehlermeldung quittiert wird:

```
Fatal error: Maximum execution time of 10 second exceeded in /var/lib/typo3/typo3_src-4.2.8/t3lib/class.t3lib_db.php on line 371
```

Sollte PHP über `mod_fcgid` eingebunden sein, ist zu beachten, dass das Apache-Modul ebenfalls ein Zeitfenster für die Verarbeitung von Skripten besitzt. In der Datei `/etc/apache2/mods-enabled/fcgid.conf` sollte somit der gleiche Wert hinter der Direktive `IPCCommTimeout` gesetzt werden.<sup>110</sup>

### 5.3.3 expose\_php

Mit der Direktive `expose_php` wird bei jeder Server-Antwort, die an den Besucher der Website geschickt wird, ein kleiner Header-Eintrag hinzugefügt, indem steht, dass

<sup>107</sup> Vgl. The PHP Group: „Beschreibung der php.ini-Direktiven des Sprachkerns“, 07.08.2009, <http://de2.php.net/manual/de/ini.core.php>, 12.08.2009 und Vgl. o.V.: „php.ini Performance Tuning“, 28.01.2009, <http://phpperformance.de/phpini-performance-tuning/>, 02.08.2009

<sup>108</sup> Vgl. The PHP Group: „PHP input/output streams“, 07.08.2009, <http://de2.php.net/manual/de/wrappers.php.php>, 02.08.2009

<sup>109</sup> Vgl. The PHP Group: „max\_execution\_time“, 07.08.2009, <http://de2.php.net/manual/de/info.configuration.php#ini.max-execution-time>, 12.08.2009

<sup>110</sup> Vgl. Stucki, Michael: „Using PHP with mod\_fcgid“, o.J., <http://typo3.org/development/articles/using-php-with-mod-fcgid/page/3/>, 01.08.2009

die Seite mit PHP generiert wurde. Dieser Eintrag ist für einen Großteil der User nicht relevant. Dennoch könnte sich ein Hacker für die Server-Konfiguration interessieren, um mit entsprechenden Mitteln Schaden zuzufügen.<sup>111</sup> Das Selbstverständnis und die Strategie eines Serveradministrators sollte es ohnehin sein, so wenige Informationen wie möglich über die verwendete Infrastruktur nach außen Preis zu geben.<sup>112</sup> Das Abschalten dieser Option spart durch den fehlenden Header-Eintrag sogar ein wenig Bandbreite.

### 5.3.4 memory\_limit

Die Direktive `memory_limit` gibt den maximal zur Verfügung stehenden Arbeitsspeicher für eine Skriptaufführung an. Ähnlich, wie die zuvor erwähnte Einstellung, dient diese als Schutz vor schlecht programmiertem Programm-Code. Wer hier bei einem komplexen Skript einen zu kleinen Wert einstellt, bekommt eine Fehlermeldung in folgender Gestalt:

```
Fatal error: Allowed memory size of 16777216 bytes exhausted (tried to allocate 12 bytes)
```

Die Direktive `memory_limit` gibt den maximal zur Verfügung stehenden Arbeitsspeicher für eine Skriptaufführung an. Auf der offiziellen TYPO3-Internetseite spricht man von mindestens 16 MB. Aus eigenen Erfahrungen sollte diese Variable für Erweiterungen, wie News-Modul auf mindestens 64 MB gesetzt.<sup>113</sup>

### 5.3.5 register\_globals

Die Option `register_globals` ist aus Sicherheitsgründen seit PHP 4.2.0 deaktiviert. Bei aktivierten `register_globals`, werden alle superglobalen, assoziativen Arrayeinträge in einer eigenen Variable abgelegt. Im Detail bedeutet dies, dass Werte, die in den Arrays `$_POST`, `$_GET`, `$_SERVER` etc. abgelegt sind, auch über eine weitere Variable im Skript aufgerufen werden können. `$_SERVER['PHP_SELF']` würde `$PHP_SELF` inhaltlich entsprechen. Benutzt ein Skript eine solche Variable, könnte der Startwert über einen übergebenen Parameter in der URL (GET), geändert werden. Dies stellt ein massives Sicherheitsrisiko dar. Auf der offiziellen Seite heißt es: „Sich auf dieses Feature zu verlassen, ist in keiner Weise empfehlenswert.“<sup>114</sup>

<sup>111</sup> Vgl. PHP Group: „Beschreibung der php.ini-Direktiven des Sprachkerns“, 07.08.2009, <http://de3.php.net/manual/de/ini.core.php>, 12.08.2009

<sup>112</sup> Vgl. o.V.: „Security by Obscurity“, o.J., <http://www.php-kurs.com/security-by-obscurity.htm>, 05.08.2009 und

Vgl. PHP Security Consortium: „PhpSecInfo Test Information - expose\_php“, o.J., [http://phpsec.org/projects/phpsecinfo/tests/expose\\_php.html](http://phpsec.org/projects/phpsecinfo/tests/expose_php.html), 02.08.2009

<sup>113</sup> Vgl. TYPO3 Association: „System Requirements“, o.J., <http://typo3.org/about/system-requirements/>, 01.08.2009

<sup>114</sup> PHP Group: „Verwendung von Register Globals“, 07.08.2009, [http://de3.php.net/manual/de/security\\_globals.php](http://de3.php.net/manual/de/security_globals.php), 01.08.2009

Dennoch ist die Einstellung bei vielen Providern aktiviert. Darüberhinaus benötigt der PHP-Interpreter für die Initialisierung von zusätzlichen Variablen Rechenleistung, die eingespart werden kann. TYPO3 benötigt kein aktiviertes `register_globals`.<sup>115</sup>

Die gezeigten Anpassungen in diesem Teil der Arbeit sind schnell in der `php.ini` vorgenommen. Obwohl aus ihnen nur eine geringe Leistungssteigerung resultiert, lohnt es sich unnötige Funktionen zu deaktivieren und damit sogar die Sicherheit des Systems zu erhöhen.

## 5.4 PHP-Beschleuniger

Bisher wurden in diesem Kapitel die verschiedenen Methoden der Integration in den Apache-HTTP-Server vorgestellt und die Konfigurationsdatei von PHP optimiert. Beides war mit den Bordmitteln von PHP und dem Apache-Webserver möglich. Der folgende Abschnitt befasst sich mit der Verbesserung der Performance durch die Verwendung von Dritt-Software. Die Rede ist von PHP-Beschleunigern. Es gibt zahlreiche Vertreter, wie z.B. der eAccelerator<sup>116</sup>, XCache<sup>117</sup> oder ZendOptimizer<sup>118</sup>. Doch bevor die verschiedenen Varianten verglichen werden, muss man zunächst verstehen, wie sie funktionieren. Man unterscheidet zwischen zwei unterschiedlichen Vorgehensweisen, die nachfolgend kurz skizziert werden sollen.<sup>119</sup>

### 5.4.1 OpCode Cache

Bei der ersten Art von PHP-Beschleunigern handelt es sich um den sogenannten OpCode Cache. Um die Funktionsweise dieser Software zu verstehen, muss man zunächst wissen, wie die Verarbeitung eines PHP-Skripts funktioniert.

Ein aufgerufenes PHP-Skript enthält Anweisungen, die ausgeführt werden sollen. Damit der Prozessor diese Anweisungen versteht, müssen sie in eine für ihn verständliche Sprache umgewandelt werden. Diese Sprache bezeichnet man als Operation Code.<sup>120</sup> Für die Umwandlung des PHP-Codes in den Operation Code ist ein Interpreter nötig, der auch als Parser bezeichnet wird. Erst nach dieser Übersetzung kann der Prozessor die Befehle umsetzen. Die Interpretation durch den PHP-Parser wird bei jedem Aufruf des PHP-Skripts erneut durchgeführt. Genau an diesem Punkt setzt der OpCode Cache an. Er speichert den vom Interpreter übersetzten Code auf der Festplatte des Servers. Beim erneuten Aufruf der PHP-Datei kann der zwischenge-

<sup>115</sup> Vgl. PHP Security Consortium: „PHP Security Guide: Overview“, o.J., <http://phpsec.org/projects/guide/1.html#1.3>, 01.08.2009 und vgl. o.V.: „php.ini Performance Tuning“, 28.01.2009, <http://phpperformance.de/phpini-performance-tuning/>, 01.08.2009

<sup>116</sup> Siehe: „eAccelerator“, o.J., <http://eaccelerator.net/>, 04.08.2009

<sup>117</sup> Siehe: „XCache“, o.J., <http://xcache.lighttpd.net/>, 19.08.2009

<sup>118</sup> Siehe: „ZendOptimizer“, o.J., <http://www.zend.com/de/products/guard/>, 19.08.2009

<sup>119</sup> Vgl. Lehr, Andreas: „PHP Bytecode Cacher im Vergleich“, 05.11.2007, <http://andreas-lehr.com/blog/archives/116-PHP-Bytecode-Cacher-im-Vergleich.html>, 04.08.2009

<sup>120</sup> Kurz: OpCode

speicherte Operation Code direkt vom Prozessor ausgeführt werden. Die aufwändige Arbeit des PHP-Parsers entfällt somit ab dem zweiten Aufruf des Skripts.<sup>121</sup>

### 5.4.2 OpCode Optimizer

Einen anderen Weg bei der Beschleunigung von PHP-Skripts, geht der OpCode Optimizer. Dieser versucht die PHP-Skripte bei der Interpretation zu optimieren. Er kennt dabei zahlreiche Routinen zur Verbesserung des Codes. Eine davon, wird durch folgendes Beispiel deutlich. Eine in PHP programmierte Schleife könnte so aussehen:

```
for ($i = 0; $i < 3; $i++) {  
    echo 'Schleife<br />';  
}
```

Der OpCode Optimizer generiert daraus folgenden Code:

```
echo 'Schleife<br />';  
echo 'Schleife<br />';  
echo 'Schleife<br />';
```

Grundsätzlich erzeugt der Code das gleiche Ergebnis. Warum ist dieser Code nun aber schneller? Zunächst müssen keine Variablen initialisiert werden, da die Zählervariable entfällt. Darüberhinaus finden keine Vergleiche statt, es muss keine Zählervariable erhöht werden und es entfallen die Sprungbefehle.<sup>122</sup>

### 5.4.3 Vergleich

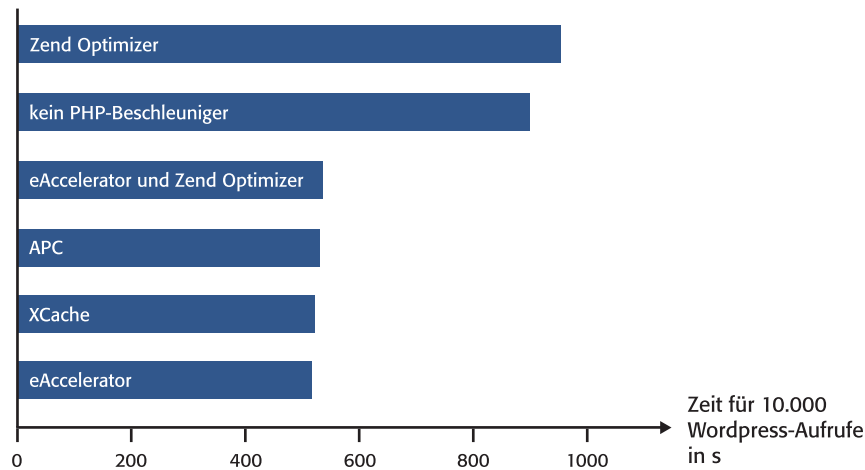
Wie bereits zu Beginn des Abschnitts beschrieben, gibt es eine Vielzahl an PHP-Beschleunigern. Es muss also der passende für das Live-System gefunden werden. Die nachfolgende Grafik zeigt die zu erwartenden Ergebnisse im Einsatz. Für die Messung wurde das auf PHP & MySQL basierende Blog-System Wordpress verwendet.<sup>123</sup> Bei diesem Test wurde 10.000 mal die Startseite des Blogs aufgerufen und die dafür benötigte Zeit gemessen.

<sup>121</sup> Vgl. Deobald, Dominik: „Benchmarking PHP: eAccelerator und andere OpCode Caches“, 11.04.2008, <http://blogs.interdose.com/dominik/2008/04/11/benchmarking-php-eaccelerator-und-andere-opcode-caches/>, 03.08.2009 und

Vgl. Ihlenfeld, Jens: „PHP Accelerator - mehr Speed für PHP“, 25.10.2001, <http://www.golem.de/0110/16571.html> 03.08.2009, 13.08.2009

<sup>122</sup> Vgl. Walter, Michael: „Voll Karacho - PHP-Beschleuniger im Vergleich“, 05/2005 <http://www.linux-magazin.de/Heft-Abo/Ausgaben/2005/05/Voll-Karacho>, 03.08.2009

<sup>123</sup> Vgl. o.V.: „About WordPress“, o.J., <http://wordpress.org/about/>, 04.08.2009

Abbildung 6: PHP-Beschleuniger im Vergleich<sup>124</sup>

Wie das Ergebnis zeigt, verbessern die eingesetzten PHP-Beschleuniger den Aufruf der Seite spürbar. Lediglich der Zend Optimizer verzögert den Seitenaufruf. Dies ist dadurch zu erklären, dass die Optimierung des Codes zunächst Rechenleistung und damit Zeit kostet. Diese benötigte Zeit wird durch den optimierten OpCode nicht aufgeholt. Theoretisch würde der Zend Optimizer in Kombination mit einem OpCode Cache besser abschneiden, da der optimierte Programmcode von der Festplatte geladen wird und somit Vorteile gegenüber den reinen OpCode Caches hätte. In der Praxis ist dieses Ergebnis jedoch nicht zu erkennen. Die Kombination aus Zend Optimizer und eAccelerator ist langsamer als alle anderen OpCode Caches im Testfeld. Als Sieger aus diesem Test geht der eAccelerator hervor. Er beschleunigt den Seitenaufruf am besten.

Erwähnt werden sollte an dieser Stelle, dass die resultierenden Aufrufzeiten von Skript zu Skript verschieden sind. Es darf jedoch davon ausgegangen werden, dass die Beschleunigung des gegenüber Wordpress komplexeren TYPO3 noch besser ausfällt. Der Einsatz eines PHP-Beschleunigers für TYPO3 ist somit zu empfehlen.

<sup>124</sup> In Anlehnung an Deobald, Dominik: „Benchmarking PHP: eAccelerator und andere OpCode Caches“, 11.04.2008, <http://blogs.interdose.com/dominik/2008/04/11/benchmarking-php-eaccelerator-und-andere-opcode-caches/>, 04.08.2009

#### 5.4.4 eAccelerator

Wie wir im vorigen Abschnitt erfahren haben, sorgt die Verwendung eines PHP-Beschleunigers für eine kürzere Laufzeit bei PHP-Skripten. Die größte Performance-Steigerung erfuhr das System durch den eAccelerator, dessen Installation und Konfiguration nun kurz beschrieben wird.

Debian besitzt kein vorkompiliertes Paket für den Einsatz in einer LAMP-Umgebung. Es muss daher von der offiziellen Website bezogen werden. Der eAccelerator steht dort in der Version 0.9.5.3 zum Download bereit.<sup>125</sup> Im nächsten Schritt muss das geladene Archiv entpackt werden. Für die Kompilierung des PHP-Beschleunigers werden drei Programme benötigt, die mit diesem Befehl über die Konsole installiert werden:

```
aptitude install gcc automake libc6-dev
```

Nachdem in das entpackte Verzeichnis gewechselt wurde, kann der eAccelerator mit den nächsten Befehlen konfiguriert, kompiliert und installiert werden:

```
export PHP_PREFIX="/usr"
$PHP_PREFIX/bin/phpize
./configure --enable-eaccelerator=shared --with-php-
    config=$PHP_PREFIX/bin/php-config
make
make install
```

Der eAccelerator wird durch die Installation direkt als PHP-Modul im richtigen Ordner abgelegt. Für die Aktivierung des Moduls müssen nun Einträge in der Konfigurationsdatei von PHP vorgenommen werden. Diese können am unteren Ende der Datei hinzugefügt werden:

```
[eAccelerator]
extension="eaccelerator.so"
eaccelerator.shm_size="32"
eaccelerator.cache_dir="/tmp"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

<sup>125</sup> Vgl. o.V.: „Release 0.9.5.3“, 15.07.2009, <http://eaccelerator.net/wiki/Release-0.9.5.3> 04.08.2009, 13.08.2009



Diese Einstellungen dienen dazu den PHP-Beschleuniger auf einem TYPO3-System zu aktivieren. Je nach Server-Umgebung sind hier individuelle Anpassungen vorzunehmen, um die maximale Leistung aus dem System herauszuholen.

Der Abschnitt hat gezeigt, wie der eAccelerator grundsätzlich zu installieren ist, um in einer TYPO3-Umgebung Verwendung zu finden. Der Performance-Gewinn ist gewaltig. Der Prozessor wird weniger belastet und kann sich somit neuen Aufgaben widmen.

## 6 MySQL

MySQL ist das populärste, frei zur Verfügung stehende Datenbankmanagementsystem. Obwohl TYPO3 mit seiner abstrahierten Programmierweise verschiedene Systeme zur Speicherung der Daten ermöglicht, wird MySQL am häufigsten genutzt. TYPO3 legt für den Betrieb zahlreiche Tabellen zur Speicherung der vielen Informationen, wie Seiteninhaltelemente, Benutzerverwaltung, Cache, etc. an. Wie bei den vorigen Ebenen, gibt es auch an der MySQL-Datenbank einige Möglichkeiten zur Optimierung, die nachfolgend näher erläutert werden.

### 6.1 Diagnose

MySQL ist ein ähnlich komplexes Thema, wie der Apache-Webserver. Welche Anpassungen für einen optimalen Produktiveinsatz nötig sind, ist abhängig von den Anforderungen und lässt sich trotz der Verwendung mit TYPO3 nicht pauschal beantworten. Bevor Maßnahmen zur Steigerung der Performance getroffen werden können, muss zunächst herausgefunden werden, was die Leistung der Datenbank limitiert. MySQL bringt eine Reihe von Werkzeugen mit, die bei der Arbeit helfen. Diese werden nun näher erläutert.

#### 6.1.1 Slow Query Log

Das erste nützliche Werkzeug nennt sich Slow Query Log und wird über die Konfigurationsdatei von MySQL aktiviert. Wie der Name vermuten lässt, werden langsame Datenbankabfragen in einer separaten Logdatei festgehalten. Grundsätzlich bieten die SQL-Abfragen ein großes Optimierungspotential, weshalb die Aufspürung langsamer Abfragen in vielen Fällen auch sinnvoll ist.<sup>126</sup>

In einer TYPO3-Umgebung ist jedoch zu bedenken, dass die Datenbankstruktur, sowie die SQL-Abfragen nicht ohne Weiteres verbessert werden können. Das CMS und seine Erweiterung basieren auf einer durchdachten und fest verankerten Struktur. Eine Veränderung könnte zu Fehlern im Produktiveinsatz führen.

Die Aufzeichnung langsamer DB-Abfragen macht dennoch Sinn. So lässt sich beispielsweise abwägen, welche Erweiterung für das Bereitstellen einer Funktionalität besser geeignet ist. Darüberhinaus kann so ermittelt werden, welche TYPO3-Seiten besonders viel Last durch die Datenbankabfrage erzeugt. Hier könnte eventuell mit dem TYPO3-internen Cache gearbeitet werden.

Die Konfigurationsdatei zum Aktivieren des Slow Query Logs ist bei Debian über folgenden Pfad zu erreichen: `/etc/mysql/my.cnf`. Mit diesen beispielhaften Einstellungen, lassen sich SQL-Befehle, die länger als 2 Sekunden zur Verarbeitung

<sup>126</sup> Vgl. Erk, Bernd: „MySQL Performance Serie – Teil 6: Slow-Query-Log“, 24.09.2008, <http://blog.netways.de/2008/09/24/mysql-performance-serie-teil-6-slow-query-log/>, 08.08.2009

benötigen in eine separate Log-Datei schreiben:

```
long_query_time = 2
log-slow-queries = /var/log/mysql-slow.log127
```

Betrachtet man diese Logdatei über einen längeren Zeitraum, lassen sich daraus problematische Abfragen erkennen. Für weitere Optionen, wie das Aufzeichnen von Administrator-Anfragen, gibt die offizielle Dokumentation weiterführende Hinweise.<sup>128</sup>

### 6.1.2 EXPLAIN

Nachdem man sich etwas genauer herangetastet hat, geht es weiter mit der Optimierung einzelner Befehle. In den meisten Fällen sorgen die langsamen `SELECT`-Befehle zur Ausgabe von Datensätzen für Kopfzerbrechen. Genau an diesem Punkt setzt der `EXPLAIN`-Befehl an. Er ist als Präfix zu verstehen und kann vor den zu analysierenden `SELECT`-Befehl gesetzt werden. Im Rahmen von TYPO3 würde solch ein Befehl so aussehen:

```
EXPLAIN SELECT DISTINCT tt_content.pid FROM tt_content
WHERE tt_content.deleted=0 ORDER BY tt_content.tstamp
DESC LIMIT 5129
```

In diesem Fall werden nicht, wie gewohnt die Datensätze der Tabelle `tt_content` angezeigt, die nicht als gelöscht markiert sind. Stattdessen gibt MySQL Informationen zurück, welche Verarbeitungsschritte nötig sind, um das Ergebnis zurückzuliefern.<sup>130</sup>

Die Interpretation des Ergebnisses erfordert eine Menge Erfahrung im Bereich MySQL. Die Dokumentation auf der offiziellen Seite gibt Aufschluss über die vielseitigen Ergebnisse, die der `EXPLAIN`-Befehl liefern kann.<sup>131</sup>

### 6.1.3 SHOW STATUS

Der anschließende MySQL-Befehl ist vergleichbar mit den im Kapitel **4.1 Diagnose** beschriebenen Modulen `mod_info` & `mod_status` für den Apache-Webserver. MySQL verrät mit diesem Befehl zahlreiche Serverinformationen, wie z.B. die Anzahl der aufgetretenen Fehler, offene Datenbankverbindungen oder den zur Verfügung stehende Speicher im Query Cache, auf den im Abschnitt **6.2.2 Query Cache** nä-

<sup>127</sup> Vgl. TYPO3 Association: „MySQL Tuning“, o.J., [http://wiki.typo3.org/index.php/Performance\\_tuning#MySQL\\_Tuning](http://wiki.typo3.org/index.php/Performance_tuning#MySQL_Tuning), 08.08.2009 und

<sup>128</sup> Vgl. Schwartz, Baron/Tkachenko, Vadim/Zaitsev, Peter et al. 2009, 69 ff. und Vgl. Kofler 2007, 530 ff. und Vgl. MySQL AB: „5.2.4. The Slow Query Log“, o.J., <http://dev.mysql.com/doc/refman/5.0/en/slow-query-log.html>, 08.08.2009

<sup>129</sup> Vgl. TYPO3 Association: „MySQL Tuning“, o.J., [http://wiki.typo3.org/index.php/Performance\\_tuning#MySQL\\_Tuning](http://wiki.typo3.org/index.php/Performance_tuning#MySQL_Tuning), 08.08.2009

<sup>130</sup> Vgl. Kofler 2007, 263 ff. und Vgl. Schwartz, Baron/Tkachenko, Vadim/Zaitsev, Peter et al. 2009, 661 ff.

<sup>131</sup> Vgl. MySQL AB: „12.3.2. EXPLAIN Syntax“, 13.08.2009, <http://dev.mysql.com/doc/refman/5.0/en/explain.html>, 08.08.2009

her eingegangen wird. Der Befehl lautet: `SHOW STATUS;` Er kann direkt über die MySQL-Konsole eingegeben werden oder über phpMyAdmin<sup>132</sup>, welches die Anfrage weiterleitet.

Diese ausführliche Liste mit Informationen, kann für die Verbesserung der MySQL-Performance genutzt werden. Sie dient als Anhaltspunkt für die nicht ganz so leichte Optimierung der Datenbank bei einer TYPO3-Installation.<sup>133</sup>

## 6.2 Konfiguration

Nachdem wir nun drei nützliche Werkzeuge zum Aufspüren eines Flaschenhalses bei MySQL kennengelernt haben, geht es weiter mit der eigentlichen Beseitigung dieser Leistungsengpässe. Dafür gibt es zahlreiche Variablen, die in der Konfigurationsdatei angepasst werden können. Bei einer Installation über Debians Paketverwaltung, sollte diese Datei über den Pfad `/etc/mysql/my.cnf` zu erreichen sein. Wichtig ist, dass nach vorgenommenen Änderungen der MySQL-Daemon neugestartet wird, damit die Einstellungen ihre Wirkung zeigen. Dies wird mit diesem Befehl über die Server-Konsole erreicht: `/etc/init.d/mysql restart`. Die Standardeinstellungen bei Debian sind für die höchstmögliche Kompatibilität vorgesehen. Sie setzen dadurch allerdings nicht besonders viel Leistung frei, wodurch Anpassungen an die eigenen Bedürfnisse vorgenommen werden müssen. Die Anpassung der einzelnen Variablen ist mit viel Arbeit verbunden. Ein Patentrezept für die Steigerung der Performance gibt es nicht.

MySQL besitzt zahlreiche Variablen, welche die Leistung von Datenbankabfragen beschleunigen können. Auf diese Vielzahl von Variablen einzugehen, würde den Rahmen dieser Arbeit sprengen. Daher werden nachfolgend die wichtigsten Variablen zur Steigerung der Performance erläutert, sodass im Anschluss das eigene TYPO3-System verbessert werden kann.

### 6.2.1 key\_buffer\_size

Laut offizieller Dokumentation gehört die Direktive `key_buffer_size` zu den Optionen, welche die Leistung am stärksten beeinflusst. Diese Variable gibt an, wie viel Arbeitsspeicher MySQL für die Speicherung von Indizes nutzt. Dieser Speicher wird benutzt, um die Schlüssel von MyISAM-Tabellen, die auch von TYPO3 genutzt werden, abzulegen. Je größer der Speicher, desto mehr Schlüssel können abgelegt werden.

<sup>132</sup> Vgl. phpMyAdmin devel team: „About“, o.J., [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php), 08.08.2009

<sup>133</sup> Vgl. MySQL AB: „13.5.4. SHOW“, o.J., <http://dev.mysql.com/doc/refman/5.1/de/show.html>, 08.08.2009, 08.08.2009

Als Faustformel gibt die Dokumentation auf der MySQL-Seite 25% des Arbeitsspeichers an. Bei einem Gigabyte Arbeitsspeicher bedeutet das einen Wert von 256M in der Konfigurationsdatei.<sup>134</sup>

Ob der Wert ausreichend für das System ist, kann mittels des in Kapitel MySQL Show-Status beschriebenen Befehl überprüft werden. Dieser Befehl liefert dabei unter anderem die Werte `Key_blocks_unused` und `Key_blocks_used`. Strebt beispielsweise `Key_blocks_unused` gegen 0, reicht die Größe des Schlüsselpuffers nicht aus.<sup>135</sup>

### 6.2.2 Query Cache

Hinter dem Begriff Query Cache verbirgt sich eine Möglichkeit zur Zwischenspeicherung von Datenbankergebnissen. Wiederkehrende Abfragen benötigen somit keine aufwändige Suche in der Datenbank, sondern greifen auf die Daten im Query Cache zurück. Dieser legt seine gespeicherten Ergebnisse im Arbeitsspeicher des Servers ab, wodurch sie schneller geladen werden.

Der Query Cache bringt nur bei auslesenden `SELECT`-Befehlen einen Geschwindigkeitsvorteil, aber auch nur dann, wenn sich die Datensätze in der Datenbank nicht ständig ändern. Dies ist bei einem TYPO3-System, trotz der Möglichkeit seine Daten zu pflegen, der Fall. Hier überwiegen die `SELECT`-Anweisungen. Eher selten kommen die `UPDATE`-, `INSERT`- und `DELETE`-Befehle zum Bearbeiten, Einfügen und Löschen vor. Somit verfallen die Daten im Zwischenspeicher nicht so schnell und die Datenkonsistenz ist gewahrt.

In der Standardkonfiguration ist der Query-Cache grundsätzlich aktiviert, aber aufgrund der Einstellung `query_cache_size 0` ist dieser quasi deaktiviert. Mit dieser Variable wird die Größe des Query Caches bestimmt. Die angegebene Speichermenge wird direkt nach dem Neustart des MySQL-Daemons im Arbeitsspeicher reserviert.<sup>136</sup>

Mit den nachfolgenden Einstellungen werden 32 MB im RAM reserviert. Darüberhinaus werden `SELECT`-Ergebnisse gespeichert, die weniger als 50 KB im Cache belegen. Damit wird dafür gesorgt, dass wenige große Ergebnisse den wertvollen Platz im Query Cache blockieren.

```
query_cache_size = 32M
query_cache_type = 1 # 0=Off, 1=On, 2=Demand
query_cache_limit = 50K137
```

<sup>134</sup> Vgl. MySQL AB: „key\_buffer\_size“, o.J., [http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar\\_key\\_buffer\\_size](http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar_key_buffer_size), 09.08.2009

<sup>135</sup> Vgl. PHP Performance: „MySQL Systemvariablen – key\_buffer\_size“, 11.08.2009, [http://phpperformance.de/mysql-systemvariablen-key\\_buffer\\_size/](http://phpperformance.de/mysql-systemvariablen-key_buffer_size/), 09.08.2009

<sup>136</sup> Vgl. Kofler 2007, S. 563 ff. und

Vgl. Schwartz, Baron/Tkachenko, Vadim/Zaitsev, Peter et al. 2009, 220 ff.

<sup>137</sup> Kofler 2007, S. 564

Abschließend sollte erwähnt werden, dass es für den Einsatz des Query Caches keine Faustformel gibt, mit der sich die Einstellungen für ein Serversystem errechnen lassen. Hier ist ein wenig Ausprobieren nötig. Der MySQL-Befehl `SHOW STATUS LIKE 'qcache%';` kann hierbei helfen. Er gibt Aufschluss über die Auslastung des Zwischenspeichers.<sup>138</sup>

### 6.2.3 table\_open\_cache

Die nächste Variable hieß vor der MySQL-Version 5.1 noch `table_cache`. Nun heißt sie `table_open_cache`. Unabhängig von der Version, verbirgt sich dahinter der Tabellen-Cache von MySQL. Für den Zugriff auf DB-Tabellen, müssen diese zunächst geöffnet werden. Nach der Operation wird die Tabelle wieder geschlossen. Beides kostet Zeit. Mit der Variable `table_open_cache` lässt sich die Anzahl der geöffneten Tabellen bestimmen. Ein hoher Wert sorgt dafür, dass viele Tabellen geöffnet bleiben. Jedoch kostet dies Speicherplatz. Mit ausreichendem Arbeitsspeicher lässt sich dieser Wert allerdings ohne Probleme erhöhen.

Um zu überprüfen, ob der Wert korrekt eingestellt ist, liefert `SHOW STATUS;` die passende Anzeige. Mithilfe der Statusvariable `open_tables` lässt sich überprüfen, ob die eingestellte Anzahl für offene Tabellen ausreicht. `table_open_cache` sollte größer als `open_tables` sein.<sup>139</sup>

<sup>138</sup> Vgl. Erk, Bernd: „MySQL Performance Serie – Teil 4: Query-Cache“, 10.09.2008, <http://blog.netways.de/2008/09/10/mysql-performance-serie-teil-query-cache/>, 10.08.2009

<sup>139</sup> Vgl. PHP Performance: „MySQL-Systemvariablen – table\_cache“, 07.08.2008, [http://phpperformance.de/mysql-systemvariablen-table\\_cache/](http://phpperformance.de/mysql-systemvariablen-table_cache/), 10.08.2009

## 7 TYPO3

Das Content Management System TYPO3 bietet einen großen Funktionsumfang, der seinen Preis hat. So werden die TYPO3-Seiten aufwändig generiert, bevor sie an den User ausgeliefert werden. Bei dieser Generierung werden Datensätze ausgelesen, Grafiken generiert und viele weitere rechenintensive Schritte durchgeführt. Nachdem in dieser Arbeit bereits die am häufigsten eingesetzten Software-Pakete zum Betreiben einer TYPO3-Installation beschleunigt wurden, wird nun das CMS an sich näher beleuchtet. Im Folgenden werden Möglichkeiten gesucht um die Serverlast zu senken und somit hochfrequentierte Webseiten auch bei Lastspitzen zügig auszuliefern. TYPO3 liefert für dieses Vorhaben einige versteckte Funktionen, die bei diesem Vorhaben helfen.

### 7.1 Caching

Wie zu Beginn des Kapitels bereits beschrieben, ist die Seitengenerierung bei TYPO3 sehr aufwändig. Um die Seite nicht bei jedem Aufruf neu zu generieren, werden Zwischenspeicher verwendet, wie man sie bereits aus den vorangegangenen Abschnitten kennt. Diese Caches ermöglichen einen schnelleren Seitenaufbau durch das Überspringen der rechenintensiven Schritte.

Bei der Zwischenspeicherung unterscheidet man grob zwischen zwei Seiten. Da gibt es zum einen die Client-Seite, wozu der User mit seinem Browser zählt. Zum anderen gibt es dann noch die bereitstellende Seite des TYPO3-Servers. Das CMS bietet für beide Seiten eine Möglichkeit zur Speicherung der bereits aufgerufenen Website-Daten.

Nachfolgend werden die nötigen Einstellungen für das Caching beider Seiten beschrieben.

#### 7.1.1 Client

Bei der Zwischenspeicherung auf Seiten des Clients, werden die Daten direkt vom lokalen Rechner des Benutzers oder von einem zwischengeschalteten Proxy, wie er häufig in Unternehmen zu finden ist, geladen. Trotz der Speicherung im Cache, ist es nötig, dass der Benutzer aktuelle Informationen erhält. Nach einer Aktualisierung des Webseiteninhalts durch den Seitenbetreiber, muss somit auch der client-seitige Cache aktualisiert werden.

Dafür gibt es in TYPO3 zwei Einstellungen, die im Setup des Seiten-Templates hinter-

legt werden müssen:

```
config.sendCacheHeaders = 1
config.sendCacheHeaders_onlyWhenLoginDeniedInBranch = 1
```

Mittels der ersten Zeile wird TYPO3 angeordnet einen speziellen HTTP-1.1-Header an den Client zu versenden. Diese für den normalen Benutzer nicht ersichtliche Information weist den Browser an, die Seite zwischenspeichern, um den erneuten Aufruf zu beschleunigen.

Die zweite Variable sorgt für den Umgang mit einem besonderen Fall, der bei der Verwendung eines passwortgeschützten Bereiches auftritt. So wird beim Aufruf einer nicht personalisierten Seite durch einen nicht eingeloggten Benutzer, diese zunächst im Zwischenspeicher eines Proxy-Servers abgelegt. Versucht nun ein eingeloggter Benutzer auf diese Seite zuzugreifen, würde er ohne die Aktivierung der zweiten Option, ebenfalls die nicht personalisierte sehen. Da dies in den seltensten Fällen gewünscht ist, sorgt die eingeschaltete Option dafür, dass die Seite erneut vom Server angefordert wird. Der Server erkennt daraufhin, dass der Benutzer eingeloggt ist und liefert die personalisierte Seite an den eingeloggten Benutzer aus.<sup>140</sup>

Die clientseitige Zwischenspeicherung der Webinhalte sorgt für eine schnelle Darstellung beim Anwender der Internetpräsenz. Darüberhinaus spart die Verwendung des Client-Caches Bandbreite, da keine Daten über die Internetleitung versendet werden. Zu bedenken ist jedoch, dass die Zugriffsstatistiken auf dem Server dadurch verfälscht werden.<sup>141</sup> Abhilfe schafft hier nur ein Aufzeichnen der Zugriffe auf Ebene des Benutzers, wie es Google Analytics<sup>142</sup> tut.

## 7.1.2 Server

Nachdem wir nun erfahren haben wie wir mit geringem Aufwand den Cache der Client-Seite nutzen, geht es nun weiter mit dem Server. Auch hier liefert TYPO3 Möglichkeiten zum serverseitigen Speichern der komplexen, zuvor aufwändig generierten, Webseiten. Darüberhinaus finden sich zahlreiche Erweiterungen im Extension Repository, welche die Caching-Funktionen von TYPO3 erweitern. Neben dem TYPO3-Cache wird im Folgenden exemplarisch eine dieser Erweiterungen erläutert.

### 7.1.2.1 Caching-Tabellen

Um die rechenintensive Erstellung von Seiten zu generieren bietet das CMS viele Cache-Mechanismen. Für einen Großteil der Daten nutzt TYPO3 dafür die Datenbank. Es ist daher nützlich zu wissen, welche Aufgabe die verschiedenen Cache-Tabellen

<sup>140</sup> Nach Höppner/Meyer/Ripfel 2008, S. 678.

<sup>141</sup> Vgl. Laborenz, Kai: „Wie sind die Cache-Control-Header gesetzt, die das Caching der Seiten in Browsern und Proxies beeinflussen?“, o.J., [http://www.typo3-handbuch.de/index.php?id=164#irfaq\\_7\\_c8854](http://www.typo3-handbuch.de/index.php?id=164#irfaq_7_c8854), 15.08.2009

<sup>142</sup> Siehe: [http://www.google.com/intl/de\\_ALL/analytics/](http://www.google.com/intl/de_ALL/analytics/)



erfüllen. Es folgt daher eine Übersicht mit den für das Caching relevanten MySQL-Tabellen und ihrer Funktion.

**Tabelle 1: Caching-Tabellen einer TYPO3-Grundinstallation<sup>143</sup>**

Tabelle	Beschreibung
cache_extension	Liste aller im TYPO3 Extension Repository erhältlichen Erweiterungen für das CMS.
cache_hash	Ablage von MD5-Hashes
cache_imagesize	Bildabmessungen, die für die Generierung der Grafik-inhalte nötig sind
cache_md5params	Mittels MD5 verschlüsselte URL-Parameter
cache_pages	Inhalte der bereits generierten Seiten
cache_pagesection	Zwischengespeicherte TypoScript-Templates
cache_typo3temp_log	Temporäre Ablage für das Rendering von skalierten Bildern, um Mehrfachbearbeitung zu vermeiden.

Die Tabelle zeigt die Caching-Tabelle einer Grundinstallation. Erweiterungen, wie *realurl* können zusätzlich weitere Tabellen für die Zwischenspeicherung anlegen.

#### 7.1.2.2 Caching im Backend beeinflussen

Außerdem ist es wichtig zu wissen, wie man das Caching von TYPO3 beeinflusst. Eventuell sollen manche Seite ja gar nicht in den Cache aufgenommen werden, da sie über dynamisch generierten Inhalt verfügen. Um das Caching-Verhalten zu steuern, gibt es grundsätzlich zwei Möglichkeiten in TYPO3. Der eine Weg erfolgt über das Backend, welches über die Seiteneigenschaften zwei nützliche Funktionen parat hält. Dieser soll zuerst beschrieben werden.

<sup>143</sup> Vgl. Höppner/Meyer/Ripfel 2008, S. 194

Abbildung 7: TYPO3-Cache im Backend beeinflussen<sup>144</sup>

Wie die **Abbildung 7** zeigt, gibt es zum einen die Möglichkeit, den Cache für eine Seite komplett zu deaktivieren. Zum anderen kann man die Zeit, nachdem die gespeicherten Inhalte automatisch verfallen, einstellen.

### 7.1.2.3 Caching mit TypoScript beeinflussen

Der andere Weg das Caching von TYPO3 zu beeinflussen, ist das Definieren spezieller Direktiven im TypoScript. Diese Direktiven lassen sich im Setup eines TYPO3-Templates hinterlegen und sind somit differenziert auf verschiedene Bereiche des Seitenbaums anwendbar. Es folgt eine Übersicht aller relevanten Direktiven.

#### config.no\_cache

Mithilfe dieser Option lässt sich das Caching deaktivieren. Ist diese Variable auf 1 gesetzt, ermöglicht es Teilbereiche der Website vom Caching auszunehmen. Dies ist in der Regel nur in der Entwicklungsphase der Website sinnvoll.

<sup>144</sup> In Anlehnung an: Höppner/Meyer/Ripfel 2008, S. 197

### config.cache\_period

Mit dieser Option lässt sich ein Verfallsdatum für den Cache definieren. Sie entspricht daher der Möglichkeit, die im Abschnitt **7.1.2.2 Caching im Backend beeinflussen**, beschrieben wird. Der zugewiesene Wert muss dabei in Sekunden angegeben werden.

### config.cache\_clearAtMidnight

Sollen die zwischengespeicherten Daten um Mitternacht verfallen, ist dies die passende Option dafür. Ist diese auf 1 gesetzt, sorgt der erste Aufruf nach Mitternacht für eine neue Generierung der Seite. Die aktualisierte Fassung befindet sich daraufhin im Cache.

### config.debug

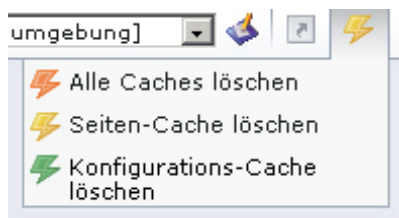
Mit der Direktive `config.debug` wird das TYPO3-Caching nicht beeinflusst. Es schaltet jedoch eine Anzeige frei, die das Ausmaß der Zwischenspeicherung beurteilen lässt. Nach dem Setzen des Wertes auf 1, ist die Anzeige im Quelltext der aufgerufenen Seite zu finden. Sie kann folgendermaßen aussehen:

```
<!-- Cached page generated 16-08-09 09:00. Expires  
17-08-09 09:00 -->  
<!-- Parsetime: 144ms -->
```

Wird die Seite nicht aus dem Cache geladen, entfällt der obere Teil der Nachricht und die Verarbeitungszeit fällt größer aus.

Mit den gezeigten Funktionen lässt sich das TYPO3-Caching an die eigenen Bedürfnisse anpassen. Je mehr Seiten aus dem Cache geladen werden, desto besser ist die Performance der TYPO3-Installation. Bei der Aktualisierung von Inhalten wird der damit korrespondierende Datensatz in der Cache-Tabelle in der Regel gelöscht. Ist dies nicht der Fall, kann der Cache natürlich immer noch separat über die Links in der rechten oberen Ecke des Backends gelöscht werden.<sup>145</sup>

### Abbildung 8: Cache im Backend löschen



<sup>145</sup> Vgl. Höppner/Meyer/Ripfel 2008, S. 197 f.

TYPO3 bietet mit den genannten Funktionen interne Lösungen um das CMS zu beschleunigen.

#### 7.1.2.4 nc\_staticfilecache

Neben den bereits beschriebenen Möglichkeiten durch TYPO3, finden sich im Online Repository eine große Anzahl an Erweiterungen zum Thema Caching. Auf alle Erweiterungen einzugehen, würde den Rahmen dieser Arbeit sprengen. Daher wird in diesem Abschnitt nur auf eine Erweiterung eingegangen. Es handelt sich um die Erweiterung `nc_staticfilecache`, die über das Online Repository in Version 2.3.1 zur Verfügung steht. Sie hat den Status „stable“, was sie für den Einsatz in einem Produktiveinsatz auszeichnet. Dies ist ein Grund, warum speziell diese Erweiterung näher beschrieben wird.

Mit `nc_staticfilecache` ist es möglich statische Dateien aus den einzelnen Seiten zu generieren. Die erzeugten Dateien werden im Dateisystem des Servers abgelegt. Dies stellt einen Unterschied gegenüber der von TYPO3 mitgelieferten Funktion dar. Dort werden die Daten in der Datenbanktabelle `cache_pages` abgelegt. Der Vorteil der statischen Dateien liegt darin, dass keinerlei Datenbankabfragen notwendig sind. Dadurch minimiert sich die Last auf dem Server. Das Besondere an `nc_staticfilecache` ist zudem, dass bei der Anfrage nach einer zwischengespeicherten Seite, TYPO3 von der Bearbeitung komplett ausgenommen wird. Dies spart wiederum wichtige Systemressourcen.

Die Extension ist in wenigen Schritten über den Erweiterungsmanager von TYPO3 installiert. Darüberhinaus benötigt die Erweiterung noch ein paar Einträge in einer Konfigurationsdatei für den Webserver. Diese Datei mit dem Namen `.htaccess`, befindet sich im Hauptverzeichnis der TYPO3-Installation. Somit wird eine Beschleunigung erzielt, wie sie mit den internen Werkzeugen von TYPO3 nicht zu erreichen wäre.

## 8 Schluss

Wie mit der vorliegenden Arbeit anschaulich geworden sein wird, steigen die Anforderungen an die Internetpräsenz tatsächlich aufgrund der Entwicklungen des Internets rasant. Das Content Management System TYPO3 bietet mit seinen frei verfügbaren Erweiterungen dafür einen großen Funktionsumfang und Möglichkeiten, damit umzugehen. Aus diesem Grund dient diese durchaus komplexe Software dennoch für viele Webseiten als Grundgerüst. Ein so umfassendes System benötigt zur Erstellung der Seiteninhalte jedoch zugleich viel Leistung. Je mehr Funktionen benötigt werden, desto mehr rechenintensive Prozesse laufen auf dem Server ab. Hat die Website zudem ein hohes Besucheraufkommen, steigt die Last des Servers weiter an. Um die Internetseite unter Lastspitzen dennoch schnell auszuliefern, sind Verbesserungen am System nötig. In der vorliegenden wissenschaftlichen Arbeit wurden daher Ansätze zur Verbesserung der Leistung für das Content Management System TYPO3 erarbeitet.

Die Maßnahmen zur Optimierung der Performance sind vielfältig. Sie verteilen sich auf unterschiedliche Ebenen, die zuvor in dieser Arbeit beschrieben wurden. Der eingangs geschilderten Problematik, dass die Webanwendungen immer ressourcenintensiver werden, kann damit entgegen gewirkt werden. Auch wenn sich diese Arbeit auf die Webapplikation TYPO3 beschränkt, können die genannten Methoden lediglich als Ansätze für das große Thema der Leistungssteigerung bei diesem CMS gesehen werden. Die Server-Konfigurationen und die darauf betriebenen Webseiten sind so individuell, dass nicht jede Maßnahme eine wirkliche Verbesserung der Leistung mit sich bringen muss.

Bei so einem komplizierten System, wie es für TYPO3 nötig ist, gibt es viele Einstellungen, die verbessert werden können. Zusammen mit der Vielzahl von Softwarepaketen, die für den Einsatz in Frage kämen, wäre das Thema zu komplex für diese wissenschaftliche Arbeit. Die Ausführungen beschränken sich daher auf die am häufigsten verwendeten Softwarepakete.

Dazu zählt der Apache-HTTP-Server. In dieser Bachelorarbeit wurde auf die grundsätzliche Funktionsweise des Webserver eingegangen. Für den Einsatz als hochfrequentierter Webserver mit größtmöglicher Skalierbarkeit bietet sich der *worker* an. Er ist ressourcensparender, als die herkömmliche Variante, die über das System angeboten wird. Um dies anschaulich zu machen, wurden die mitgelieferten Diagnose-Werkzeuge vorgestellt. Dies ist wichtig, um herauszufinden welcher Teil der Software die Auslieferung der Website ausbremst. Mit den vermittelten Grundlagen über die Diagnoseprogramme, lassen sich auch Probleme in anderen Anwendungen erkennen.

Die Skriptsprache PHP wird von TYPO3 vorausgesetzt. Von den Entwicklern wird sie selber als „Klebstoff“ mit vielen Funktionen bezeichnet. Zu Beginn des Kapi-

tels wurde zunächst die grundlegende Verarbeitung eines PHP-Skripts beschrieben. Dies war nötig, um die späteren Ansätze besser zu verstehen.

Wichtig für die Interpretation der Skripte ist die Integration in den Apache-Webserver. Es wurden die verschiedenen Varianten erläutert und die beste für den Einsatz in TYPO3 ermittelt. Darüberhinaus ermöglicht auch die Konfigurationsdatei, unabhängig von der Implementierung des Interpreters weitere kleine Verbesserung. Die größte Leistungssteigerung ist jedoch durch einen externen PHP-Beschleuniger zu erwarten. Er integriert sich in die Verarbeitung des PHP-Skripts und speichert bereits interpretierte Skripte auf der Festplatte.

Auf der Festplatte speichert TYPO3 zudem auch die Inhalte der Website. Für einen Großteil der Daten verwendet das CMS dafür eine MySQL-Datenbank. Die in Datenbank-Tabellen abgelegten Daten werden aufwändig ausgelesen, um dann in das Websitegerüst gepackt zu werden. Auch dieser Vorgang kostet Ressourcen. Das komplizierte Datenbanksystem bietet auch dafür zahlreiche Optionen zur Verbesserung. Zusammen mit den internen Diagnose-Werkzeugen, lasse sich so weitere Leistungsreserven freisetzen.

TYPO3 bietet mit dem Zwischenspeichern von bereits erstellten Seiten ebenfalls eine hervorragende Möglichkeit die Auslieferung der Seite zu beschleunigen. Dafür müssen die bereitgestellten Variablen optimal an die eigenen Bedürfnisse angepasst werden. Da es auch hier keine Patentlösung gibt, werden die relevanten Direktiven systematisch erklärt.

Abschließend lässt sich sagen, dass es möglich ist TYPO3 für den Einsatz bei einer hochfrequentierten Webseite zu nutzen. Die eingangs formulierte Aufgabe wurde damit erfüllt. Ein Leistungsgewinn lässt sich auch ohne den Kauf besserer Hardware realisieren. Dies spart enorme Kosten. Darüberhinaus ist in dieser Arbeit deutlich geworden, wie komplex eine solche Software-Konfiguration ist. Nicht jede der beschriebenen Ansätze verbessert die Performance auf allen Systemen. Es ist daher wichtig mit den beschriebenen Diagnoseprogrammen umzugehen und die gelieferten Ergebnisse korrekt zu interpretieren.

## 9 Literaturverzeichnis

### 9.1 Bücher

*Höppner, Irene/Mey, Melanie/Ripfel, Franz:* Das TYPO3 Profihandbuch. Der Leitfaden für Entwickler und Administration zu Version 4.1., München 2008

*Hücükyılmaz, Hakan/Haas, Thomas M./Merz, Alexander:* Einsteigen und durchstarten mit PHP 5. Grundlagen, Objektorientierung und PEAR. 1. Auflage, Heidelberg 2005

*Kannengiesser, Matthias:* Objektorientierte Programmierung mit PHP 5., Kevelaer 2007

*Kersken, Sascha:* Apache 2.2. Das umfassende Handbuch. 3. Auflage, Bonn 2009  
*Kofler, Michael:* MySQL 5. Einführung, Programmierung, Referenz. 3. Auflage, München 2007

*Schmidt, Stephan:* PHP Design Patterns. 2. Auflage, München 2007

*Schwartz, Baron/Tkachenko, Vadim/Zaitsev, Peter et al.:* High Performance MySQL. Optimierung, Datensicherung, Replikation & Lastverteilung. 2. Auflage, Köln 2009

*Stöckl, Andreas:* Content Management mit TYPO3. 2. Auflage, Bonn 2004

### 9.2 Zeitschriften

*Baschny, Ernesto:* Schnell, schneller, am schnellsten. TYPO3 Performance für jedermann.  
In: t3n – Open Source & Web, 03/2009, S. 123 ff.

*Bold, Max:* Auswahlkriterien: Provider-Angebote. Der richtige Provider.  
In: PHP-Journal, 07/2009, S. 112 ff.

*Bold, Max:* Optimierungsmöglichkeiten bei V-Servern. Ressourcen schonen.  
In: PHP-Journal, 07/2008, S. 116 ff.

*Egle, Christian:* Eigenbetrieb, Colocation, oder Managed Hosting. Qual der Wahl.  
In: PHP-Journal, 07/2008, S. 116 ff.

*Hauser, Tobias/Neufeind, Stefan:* Hosting für Profis. Tipps zur Provider-Wahl bei professionellem Server-Hosting.  
In: t3n – Open Source & Web, 12/2008, S. 43 ff.

*Kolyshkin, Kirill:* Virtualisierung hat viele Facetten. Emulation, Paravirtualisierung, Betriebssystem-Virtualisierung.  
In: t3n – Open Source & Web, 03/2008, S. 65 ff.

*Neufeind, Stefan*: Volle Kontrolle. Tipps zur Hoster-Wahl bei Root Servern.  
In: t3n – Open Source & Web, 06/2009, S. 51 ff.

### 9.3 Internetquellen

*ALL-INKL.COM*: Webhosting, o.J.,  
<http://all-inkl.com/index.php?open=uebersicht&sek=webhosting>, 20.07.2009

*Apache Friends*: Willkommen bei Apache Friends,  
<http://www.apachefriends.org/de/index.html>, 03.08.2009

*Apache Software Foundation*: Apache HTTP Server Project, o.J.,  
<http://httpd.apache.org/>, 12.08.2009

*Apache Software Foundation*: Apache Module mod\_info, o.J.  
[http://httpd.apache.org/docs/2.2/mod/mod\\_info.html](http://httpd.apache.org/docs/2.2/mod/mod_info.html), 31.07.2009

*Apache Software Foundation*: Apache Module mod\_deflate, o.J.,  
[http://httpd.apache.org/docs/2.2/mod/mod\\_deflate.html](http://httpd.apache.org/docs/2.2/mod/mod_deflate.html), 23.07.2009

*Apache Software Foundation*: Apache-MPM prefork, o.J.  
<http://httpd.apache.org/docs/2.2/mod/prefork.html>, 12.08.2009

*Apache Software Foundation*: Authentication, Authorization and Access Control,  
o.J., <http://httpd.apache.org/docs/2.2/howto/auth.html>, 31.07.2009

*Apache Software Foundation*: Downloading the Apache HTTP Server,  
<http://httpd.apache.org/download.cgi>, 05.08.2009

*Apache Software Foundation*: Dynamic Shared Object (DSO) Support, o.J.,  
<http://httpd.apache.org/docs/2.0/dso.html>, 22.07.2009

*Apache Software Foundation*: HostnameLookups-Direktive, o.J.,  
<http://httpd.apache.org/docs/2.0/mod/core.html#hostnamelookups>, 25.07.2009

*Apache Software Foundation*: Multi-Processing-Module (MPMs), o.J.,  
<http://httpd.apache.org/docs/2.2/mpm.html>, 12.08.2009

*Apache Software Foundation*: Sample Configurations, o.J.,  
[http://httpd.apache.org/docs/2.2/mod/mod\\_deflate.html#recommended](http://httpd.apache.org/docs/2.2/mod/mod_deflate.html#recommended),  
24.07.2009

*Apache Software Foundation*: What ist he Apache http Server Project?, o.J.,  
[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html), 11.08.2009

*Buytaert, Dries*: About Drupal, 03.07.2009,  
<http://drupal.org/about>, 08.07.2009

*Cheng, Alex/Evans, Mark*: Inside Twitter – An In-Depth Look Inside the Twitter  
World, 06/2009,  
<http://www.sysomos.com/insidetwitter/>, 01.07.2009



- Client-Server, 09.08.2007,  
<http://www.e-teaching.org/technik/vernetzung/architektur/client-server/>, 3.07.2009
- Debian Projekt*: Paket: apache2 (2.2.9-10+lenny4), o.J.,  
<http://packages.debian.org/lenny/apache2>, 05.08.2009
- Debian Projekt*: Paket: php5 (5.2.6.dfsg.1-1+lenny3), o.J.,  
<http://packages.debian.org/lenny/php5>, 27.07.2009
- Deobald, Dominik*: „Benchmarking PHP: eAccelerator und andere OpCode Caches“, 11.04.2008,  
<http://blogs.interdose.com/dominik/2008/04/11/benchmarking-php-eaccelerator-und-andere-opcode-caches/>, 03.08.2009
- Diedrich, Oliver*: Trendstudie Open Source – Eingesetzte Produkte, 04.02.2009  
<http://www.heise.de/open/Trendstudie-Open-Source-/artikel/126682/8>, 11.08.2009
- Dornfest, Rael*: LAMP Lighter: The Apache Toolbox, 17.11.2000,  
<http://onlamp.com/pub/a/apache/2000/11/17/wrangler.html> 02.08.2009
- Eicker, Thomas*: Layout Engine, o.J.,  
<http://www.kleines-lexikon.de/w/l/layoutengine.shtml>, 04.07.2009
- Erk, Bernd*: MySQL Performance Serie – Teil 6: Slow-Query-Log, 24.09.2008,  
<http://blog.netways.de/2008/09/24/mysql-performance-serie-teil-6-slow-query-log/>, 08.08.2009
- Free Software Foundation*: GNU General Public License, 29.06.2007,  
<http://www.gnu.org/copyleft/gpl.html> 08.07.2009
- heise Developer*: Benchmark of PHP 5 Versions, 30.06.2009,  
<http://www.heise.de/developer/Was-aendert-sich-mit-PHP-5-3-/zoom/140003/2>, 19.08.2009
- Heise Zeitschriften Verlag GmbH & Co. KG*: heise Security“, o.J.,  
<http://www.heise.de/security/>, 19.07.2009
- Hostloco.com*: Häufige Fragen, o.J., [http://www.hostloco.com/fragen\\_de.phtml](http://www.hostloco.com/fragen_de.phtml), 19.07.2009
- Ihlenfeld, Jens*: „PHP Accelerator - mehr Speed für PHP“, 25.10.2001,  
<http://www.golem.de/0110/16571.html> 03.08.2009, 13.08.2009
- iX – Magazin für professionelle Informationstechnik*: Server-Housing, o.J.,  
<http://www.heise.de/ix/server-housing/>, 12.08.2009
- Joomla Team*: What is Joomla?, o.J.,  
<http://www.joomla.org/about-joomla.html>, 11.08.2009
- Laborenz, Kai*: „Wie sind die Cache-Control-Header gesetzt, die das Caching der Seiten in Browsern und Proxies beeinflussen?“, o.J., [http://www.typo3-handbuch.de/index.php?id=164#irfaq\\_7\\_c8854](http://www.typo3-handbuch.de/index.php?id=164#irfaq_7_c8854), 15.08.2009

*Landschoof, Axel/Ostertag, Markus/Tyka, Jürgen*: „Analyse von Systemperformanz“, o.J.,  
<http://www2.net.in.tum.de/teaching/WS06/performprak/ausarbeitungen/Webserver-Auswertung.pdf>, 29.07.2009

*Leemhuis, Thorsten*: Kernel-Log: 2.6.26-Entwicklung läuft schwungvoll an, Fehlerkorrektur für 2.6.24 und 2.4.36, 21.04.2008,  
<http://www.heise.de/newsticker/Kernel-Log-2-6-26-Entwicklung-laeuft-schwungvoll-an-Fehlerkorrekturen-fuer-2-6-24-und-2-4-36-/meldung/106755>, 19.07.2009

*Lehr, Andreas*: „PHP Bytecode Cacher im Vergleich“, 05.11.2007,  
<http://andreas-lehr.com/blog/archives/116-PHP-Bytecode-Cacher-im-Vergleich.html>, 04.08.2009

*Mansmann, Urs*: Kanzlerin will Breitbandausbau forcieren, 01.02.2009,  
<http://www.heise.de/newsticker/Kanzlerin-will-Breitbandausbau-forcieren-/meldung/126696>, 02.07.2009

*Marsching, Sebastian*: suPHP, 14.03.2009,  
<http://www.suphp.org/Home.html>, 28.07.2009

*Microsoft*: Internet Information Services, o.J.,  
<http://www.iis.net/>, 05.08.2009

*MySQL AB*: 5.2.4. The Slow Query Log, o.J.,  
<http://dev.mysql.com/doc/refman/5.0/en/slow-query-log.html>, 08.08.2009

*MySQL AB*: 12.3.2. EXPLAIN Syntax, 13.08.2009,  
<http://dev.mysql.com/doc/refman/5.0/en/explain.html>, 08.08.2009

*MySQL AB*: 13.5.4. SHOW, o.J.,  
<http://dev.mysql.com/doc/refman/5.1/de/show.html> 08.08.2009, 08.08.2009

*MySQL AB*: key\_buffer\_size, o.J., [http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar\\_key\\_buffer\\_size](http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar_key_buffer_size), 09.08.2009

*MySQL AB*: MySQL AB Completes Record Year, 30.01.2007  
[http://www.mysql.com/news-and-events/generate-article.php?id=2007\\_02](http://www.mysql.com/news-and-events/generate-article.php?id=2007_02), 18.07.2009

*MySQL AB*: Warum MySQL?, o.J.,  
<http://www.mysql.de/why-mysql/>, 03.08.2009

*Netcraft*: July 2009 Web Server Survey, 28.07.2009,  
[http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html), 12.08.2009

*nixCraft*: Howto: Performance Benchmarks a Webserver, o.J.,  
<http://www.cyberciti.biz/tips/howto-performance-benchmarks-a-web-server.html>, 07.08.2009

*nixCraft*: Lighttpd PHP fastcgi configuration, o.J.,  
<http://www.cyberciti.biz/tips/lighttpd-php-fastcgi-configuration.html>, 30.07.2009

*Nix, Markus:* Was ist eigentlich Open Source?, 01.01.2005,  
[http://www.contentmanager.de/magazin/artikel\\_843\\_was\\_ist\\_eigentlich\\_open\\_source.html](http://www.contentmanager.de/magazin/artikel_843_was_ist_eigentlich_open_source.html), 11.08.2009

*ORACLE Deutschland:* Warum Oracle?, o.J.,  
<http://www.oracle.com/lang/de/database/index.html>, 03.08.2009

o.V.: „About WordPress“, o.J.,  
<http://wordpress.org/about/>, 04.08.2009

o.V.: *Apache: php4 und php5 parallel*,  
<http://admirableadmin.de/32/apache-php4-und-php5-parallel>, 28.07.2009

o.V.: *Apache2/workerMPM/FastCGI/PHP5*, 04.09.2008,  
<http://www.digitalnerds.net/featured/apache2-worker-mpm-with-fastcgi-php5/>, 12.08.2009

o.V.: „Memory usage Apache + PHP as module versus FastCGI“, 23.05.2009,  
<http://www.apachelounge.com/viewtopic.php?p=10991>, 30.07.2009

o.V.: *mod\_php vs. PHP-CGI*, 08.07.2009  
[http://wiki.rootforum.de/scripting/php/mod\\_php\\_vs\\_php-cgi](http://wiki.rootforum.de/scripting/php/mod_php_vs_php-cgi), 27.07.2009

o.V.: „php.ini Performance Tuning“, 28.01.2009,  
<http://phppperformance.de/phpini-performance-tuning/>, 02.08.2009

o.V.: „Release 0.9.5.3“, 15.07.2009,  
<http://eaccelerator.net/wiki/Release-0.9.5.3>, 04.08.2009

o.V.: „Security by Obscurity“, o.J., <http://www.php-kurs.com/security-by-obscurity.htm>, 05.08.2009

o.V.: *Webserver Lighttpd unter Debian 5.0 „Lenny“ Howto*, 17.06.2009,  
<http://debian.asconix.com/lighttpd-webserver-debian-lenny-howto>, 06.08.2009

*PostgreSQL Global Development Group:* About, o.J.  
<http://www.postgresql.org/about/>, 03.08.2009

*phpMyAdmin devel team:* „About“, o.J.,  
[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php), 08.08.2009

*PHP Performance:* MySQL Systemvariablen – *key\_buffer\_size*, 11.08.2009,  
[http://phppperformance.de/mysql-systemvariablen-key\\_buffer\\_size/](http://phppperformance.de/mysql-systemvariablen-key_buffer_size/), 09.08.2009

*PHP Performance:* MySQL-Systemvariablen – *table\_cache*, 07.08.2008,  
[http://phppperformance.de/mysql-systemvariablen-table\\_cache/](http://phppperformance.de/mysql-systemvariablen-table_cache/), 10.08.2009

*PHP Security Consortium:* „PHP Security Guide: Overview“, o.J.,  
<http://phpsec.org/projects/guide/1.html#1.3>, 01.08.2009

*PHP Security Consortium:* „PhpSecInfo Test Information - *expose\_php*“, o.J.,  
[http://phpsec.org/projects/phpsecinfo/tests/expose\\_php.html](http://phpsec.org/projects/phpsecinfo/tests/expose_php.html), 02.08.2009

*Pro-Linux*: ps – Prozess-Status, 22.11.1999 [http://www.pro-linux.de/t\\_shell/ps.html](http://www.pro-linux.de/t_shell/ps.html), 12.08.2009

Roos, Michiel: „Static File Cache“, 2007, [http://typo3.org/documentation/document-library/extension-manuals/nc\\_staticfilecache/2.3.1/view/1/1/#id3975723](http://typo3.org/documentation/document-library/extension-manuals/nc_staticfilecache/2.3.1/view/1/1/#id3975723), 17.08.2009

Ryan, Dominic: Difference between PHP thread safe and non thread safe binaries, 27.09.2007, [http://www.iis-id.com/articles/my\\_word/difference\\_between\\_php\\_thread\\_safe\\_and\\_non\\_thread\\_safe\\_binaries](http://www.iis-id.com/articles/my_word/difference_between_php_thread_safe_and_non_thread_safe_binaries), 28.07.2009

Schäfer, Mathias/Strübig, Joachim: Einführung in JavaScript und DOM, o.J., <http://de.selfhtml.org/javascript/intro.htm>, 03.07.2009

Schaumann, Philip: Neue Social Networks und neue Formen der Kollaboration, o.J., <http://philipps-welt.info/social%20networks.htm>, 03.07.2009

Silva, Gustavo Noronha: APT HOWTO, 04/2003, <http://www.debian.org/doc/manuals/apt-howto/index.de.html>, 19.08.2009

Stucki, Michael: Leaving PHP4 behind... 13.07.2007, <http://buzz.typo3.org/people/stucki/article/leaving-php4-behind/>, 18.07.2009

Stucki, Michael: Using PHP width mod\_fcgid, <http://typo3.org/development/articles/using-php-with-mod-fcgid/page/2/>, 29.07.2009

The Perl Foundation: About Perl, o.J., <http://www.perl.org/about.html>, 03.08.2009

The PHP Group: „Beschreibung der php.ini-Direktiven des Sprachkerns“, 07.08.2009, <http://de2.php.net/manual/de/ini.core.php>, 12.08.2009

The PHP Group: Installation, 21.07.2009, <http://de3.php.net/manual/de/faq.installation.php#faq.installation.apache2>, 26.07.2009

The PHP Group: „max\_execution\_time“, 07.08.2009, <http://de2.php.net/manual/de/info.configuration.php#ini.max-execution-time>, 12.08.2009

The PHP Group: „PHP input/output streams“, 07.08.2009, <http://de2.php.net/manual/de/wrappers.php.php>, 02.08.2009

PHP Group: „Verwendung von Register Globals“, 07.08.2009, <http://de3.php.net/manual/de/security.globals.php>, 01.08.2009

The PHP Group: What is PHP?, o.J., <http://www.php.net/>, 03.08.2009

*Timme, Falko*: Bandbreite sparen mit Apache2s mod\_deflate, 15.05.2009,  
[http://www.howtoforge.de/howto/bandbreite-sparen-mit-apache2s-mod\\_deflate/](http://www.howtoforge.de/howto/bandbreite-sparen-mit-apache2s-mod_deflate/),  
24.07.2009

*Tim O'Reilly*: What is Web 2.0, 30.09.2005,  
<http://oreilly.com/web2/archive/what-is-web-20.html>, 01.07.2009

*TYPO3 Association*: MySQL Tuning, o.J.,  
[http://wiki.typo3.org/index.php/Performance\\_tuning#MySQL\\_Tuning](http://wiki.typo3.org/index.php/Performance_tuning#MySQL_Tuning), 08.08.2009

*TYPO3 Association*: Online Repository:  
<http://typo3.org/extensions/repository/>, 22.08.2009

*TYPO3 Association*: System Requirements, o.J.,  
<http://typo3.org/about/system-requirements/>, 18.07.2009

*TYPO3 Association*: TYPO3 Association, o.J.,  
<http://typo3.de/Facts-and-Figures.factsandfigures.0.html>, 09.07.2009

*TYPO3 Association*: TYPO3 Download, o.J.,  
<http://typo3.org/download/packages/>, 04.08.2009

*Walter, Michael*: „Voll Karacho - PHP-Beschleuniger im Vergleich“, 05/2005,  
<http://www.linux-magazin.de/Heft-Abo/Ausgaben/2005/05/Voll-Karacho>,  
03.08.2009

*Walther, Jan*: Fast-CGI – was steckt dahinter?, 21.05.2007,  
<http://phpperformance.de/fast-cgi-was-steckt-dahinter/>, 01.08.2009

*Weis, Holger*: nc\_staticfilecache mit RealURL & .html-Suffix, 22.12.2009,  
[http://www.betawax.de/blog/2008/12/22/nc\\_staticfilecache-mit-realurl-html-suf-](http://www.betawax.de/blog/2008/12/22/nc_staticfilecache-mit-realurl-html-suffix/)  
[fix/](http://www.betawax.de/blog/2008/12/22/nc_staticfilecache-mit-realurl-html-suffix/), 19.08.2009

*Yahoo Developer Network*: Gzip Components, o.J.,  
<http://developer.yahoo.com/performance/rules.html#gzip>, 23.07.2009

## 9.4 Internetquellen (Wikipedia)

Clean URLs, [http://de.wikipedia.org/wiki/Clean\\_URLs](http://de.wikipedia.org/wiki/Clean_URLs), 23.07.2009

Datenübertragungsrate, <http://de.wikipedia.org/wiki/Datenübertragungsrate>,  
19.07.2009

FastCGI, <http://de.wikipedia.org/wiki/FastCGI#Funktionsweise>, 29.07.2009

Objektorientierte Programmierung, [http://de.wikipedia.org/wiki/Objektorientier-](http://de.wikipedia.org/wiki/Objektorientierte_Programmierung)  
[te\\_Programmierung](http://de.wikipedia.org/wiki/Objektorientierte_Programmierung), 14.08.2009

PHP-Funktionsweise, 01.07.2009, [http://de.wikipedia.org/wiki/Datei:PHP\\_funktions-](http://de.wikipedia.org/wiki/Datei:PHP_funktionsweise.svg)  
[weise.svg](http://de.wikipedia.org/wiki/Datei:PHP_funktionsweise.svg), 19.08.2009

Refactoring, <http://de.wikipedia.org/wiki/Refactoring>, 26.07.2009

## 10 Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht. Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.